TABLE OF CONTENTS

APPENDIX		Page
A	Timer Board	98
B	UART Description	104
C	PROM Board	114
D	Test Outline Result	121
E	Flow Diagrams	129
F	Navigation System Error Calculation	142

ABSTRACT

This report summarizes two years of work under Contract No. N66001-79-C-0055 with the Naval Ocean Systems Center, in support of the Research and Development Program, OCS Oil and Gas Operations, U.S. Geological Survey. It describes the engineering development of a microprocessor controlled, unmannedd, free swimming vehicle called EAVE (Experimental Autonomous Vehicle - East), designed as a test bed for the development of technology relating to the inspection of underwater pipelines and structures.

This report describes system developments in vehicle design, computer and software evolvement, navigation and communication system developments, through to the integration of software and hardware into an adaptive untethered free swimming vehicle which in this period, succeeded in acquiring and following an exposed underwater pipeline.

This report also summarizes work done under Contract No. N66001-80-C-0050 entitled "Navigation Concept Development for Autonomous Vehicle System". This contract relies on the support of the EAVE vehicle and its testing has been integral with the development of the vehicle itself. The principle consequences of this contract are described in Section 4 of this report.

Attention is drawn to the Program Development Plan fitting this years activity with the Programs objectives.

PREFACE

The purpose of the EAVE program is to develop technology for unmanned, free-swimming vehicles capable of performing inspection tasks on underwater pipelines and offshore structures. It is an attempt to be quite independent of man - hence the name Experimental Autonomous Vehicle (EAVE). The program is not vehicle-development, nor is it the optimization of the various subsystems for unmanned submersibles. Instead, it is investigation of technologies existing in the scientific and technological communities for the purpose of establishing and demonstrating new ways of performing basic underwater tasks of potential importance to the Geological Survey's offshore regulatory mission. Underwater pipelines and structural inspection tasks are of direct concern to the U.S. Geological Survey's (USGS's) Research and Development Program, whose major objectives are to develop technology for pollution prevention and safety in outer continental shelf oil and gas operations.

Underwater systems employing divers were the earliest means of inspecting objects and performing other useful work. Over the years, these systems have used nonsaturation and, for deeper structures, saturation diving techniques in which the diver's blood was either nonsaturated or saturated with dissolved gasses.

Developments in one atmosphere manned submersibles for inspection and other non-military purposes began in the early

1930's, and were exemplified by the Beebe-Barton bathysphere. The usefulness of these early submersibles was limited to observations down to about 4000 feet, their occupants being protected from ambient sea pressure by heavy pressure hulls. The desire to make these vehicles competitive with divers in work effectiveness led to developments of "free-swimming" manned submersibles beginning in the early 1960's.

The evolution of these vehicles has led in turn to two types of submersibles in current use - manned and untethered such as ALVIN, and unmanned but tethered such as SCORPIO. Manned tethered vehicles have been explored and built in at least one case. All these systems require a human operator to control the vehicle, whether on board the submersible or remotely through a tether cable.

It can be seen that the classification of the various unmanned vehicle systems is based upon the support provided by the surface craft. A towed system's cable, for example, transmits mechanical power to the vehicle as well as electrical power and communications. A tethered submersible cable transmits only electrical power and communications support, permitting independant motion of the vehicle. A totally autonomous vehicle is independant of support for the mission duration. Progression toward the use of totally autonomous vehicles reduces, substantially, the requirements for surface ship support.

The feasibility of attaining future advances in unmanned autonomous vehicles is based upon developments in electronic and computer micro-miniaturization, associated software and low-power, large-scale integrated (LSI) circuits which permit use of preprogrammed electronic feedback control systems.

No field of technology is more dynamic than that of the microprocessor, and its support chips. The microcomputer technology, key to unmanned autonomous vehicle development, is expanding dramatically and the decision-making logic is readily available to make more and more complex decisions. It is this technology that permits design of a level of intelligence into the prototype, free-swimming vehicle.

One consequence of putting this intelligence into a robot vehicle is the potential for making the vehicle adaptive, i.e., to allow the one platform to serve a variety of objectives. The prototype EAVE vehicle contains a first generation program for pipeline following while systems are in design to permit it to transit and inspect 3-dimensional structures.

A second consequence is the ability to employ on-board sensors to derive status information that permits the vehicle to adapt and to cope with its environment.

The removal of the tether, made possible by the on-board microprocessor, has pointed to the future use of autonomous vehicles. Under the best of circumstances, however, the autonomous vehicle must remain an extension of the intellegence of its operators, for the mission parameters always must be controlled and performance monitored, work accomplished and intelligence coveyed. Devoid of a hard-wire link, in the absense of more exotic alternatives, we must now rely for communications on acoustic telemetry, low in bandwidth and reliability, and plagued with mission related hazards. Fortunately, the microprocessor, while permitting the removal of the tether, has reduced substantially the demands that must be placed on that link. The operator now oversees the vehicle's task -- he does not have to "fly" the vehicle, and overseeing takes less bandwidth and tolerates more noise than "flying". The vehicle system now will also have inputs from various sensors that allow it to understand its surroundings and navigate within its work area. Principal among these systems is the sensor that permits the vehicle to locate itself in a geodetic grid and in precise relation to work stations and obstacles. On-board navigation skills are essential to a successful autonomous vehicle. It is such advances as these, intended to extend man's effectiveness at reduced cost and danger, which are being pursued in the Experimental Autonomous Vehicle (EAVE) program.

1. Background

The engineering design and development of a microcomputer-controlled unmanned free-swimming vehicle is being accomplished at the University of New Hampshire in conjunction with the Naval Ocean Systems Center, San Diego, under NOSC Contract No. N66001-79-C-0055. A second, directly related contract through the same sponsorship, Contract No. N66001-80-C-0050, relates to the development of a short range navigation system for the vehicle. This submersible currently is designed to follow an exposed underwater pipeline. This design objective requires that the vehicle possess a degree of intelligence which permits it to sense its environment and to make, as well as execute, control decisions.

It is increasingly clear that rapidly evolving microcomputer technology, as well as the decreasing cost of semiconductor memory is dramatically increasing the implicit capability of an autonomous vehicle to accomplish tasks of greatly increased sophistication.

The existance of an intelligent underwater autonomous vehicle, with capacity for growth, generates two generic sets of needs:

a) requirements for new technologies to support the missions of an autonomous vehicle

b) requirements for improved functional operating capability with the competence to perform new missions

This year's efforts have focused around four rather specific goals. Two of these goals involve investigations into the application of new technology to the vehicle system. The first goal was to assess the perceived needs for an autonomous vehicle navigation system. Having defined the needs, a system was chosen, fabricated and readied for evaluation tests. The second goal was to assess the perceived needs for communicating with an autonomous vehicle system. A specific low frequency data link would be defined and fabricated for integration into the vehicle system. The link was then to be used in a series of evaluation tests.

The remaining two goals dealt with the operational capability of the vehicle system. The first of these two goals was to modify the vehicle computer system, both hardware and software, to upgrade its capability and to modify the platform structure to eliminate some of the design limitations of the original platform. Specifically, three new modules were added to the computer system: a "Time" module which allows computer controlled timing to an accuracy of one part in ten million: a "PROM" module which allows the use of semi-permanent memory: A "UART" module which adds greatly increased communication capabilities. As a second operational goal, the platform battery system was changed to incorporate sealed lead acid gel cells in a pressure container, and the flotation system was changed to a more reliable configuration.

1.A CONCLUSIONS AND RECOMMENDATIONS

Summation

The work of the two years covered by this report resulted in:

1. a vehicle capable of serving as a test bed for a wide range of related technologies.

2. a demonstration that a relatively simple vehicle system can indeed follow an exposed pipeline, employing onboard sensors and resident software programs.

3. demonstration of a vehicle tracking system, which also is capable of serving as a precision navigation system.

4. demonstration of a methodology for vehicle homing, designed to support reliable recovery of an operational vehicle.

5. the development of a fundamental understanding of the implications and boundaries in the use of microprocessors in an underwater vehicle system.

6. construction of a working computer that serves as a vehicle's guidance and control intelligence, with the

ability to accept data, to make decisions, and to execute the required actions.

7. development of essential system software that enables autonomous performance.

8. design of a short range acoustic communication system to permit supervisory control and the exchange of status information with a remote observer.

9. first level assessment of required communication functions in terms of permissible error rates, bandwith allocation, and the requirements for redundnacy encoding.

LEARNINGS

The program has demonstrated that:

1. Autonomons, employing a microprocessor supervisory and control system, have substantial promise for a relatively wide range of underwater inspection tasks.

2. The vehicle system need not be very expensive or complicated to perform complex maneuvers under software control.

3. Power systems are now available which permit inspection missions of substantial duration.

4. Competent navigation and communications are essential elements in an effective autonomous vehicle mission.

RECOMMENDATIONS

Recommendations growing out of the learnings of this program include:

1. The UNH vehicle with its major control in four degrees of freedom, and minor control in one more, should be addressed to problems that are three-dimensional in scope, such as the inspection of structures. The pipeline inspection task, as a primary system objective should be delegated to a 3-degree of freedom vehicle.

2. The movement of the vehicle with great precision through a structure places substantial demands on the navigation system. Attention must be placed on a design that achieves probable errors of only a few inches, and that solves the evident problems of shadowing and multipath.

3. Further attention should be placed on relatively short range, wide bandwidth communication systems, thus allowing real time displays and eventually television to be available for a remote observer.

4. Power systems have been identified that are more effective than the currently-used lead acid batteries. Application of these new cells to EAVE should be considered.

5. A structural inspection mission will involve substantial amounts of program, which will probably overburden the 6100 microprocessor employed in the present vehicle. The demands of such missions require that early effort be placed on choosing a competent microprocessor, and on designing the required hardware and software systems.

6. Attention must be given to the sensors, as well as the tools for inspection, that are appropriate to the structural inspection mission. Although it is not desirable that such developments be accomplished in the EAVE program, it is necessary to address the problem of interfacing with the available sensors and tools needed at the work station.

2. The Platform

The "torpedo" unmanned, untethered submersible has been in existence since the early days of submarine warfare, being used as a weapon employing progressively sophisticated, on-board systems control to determine its depth, course and warhead activation. In recent years, "torpedo" vehicles applicable for search and inspection missions under ice, as well as in open water, have been developed by the Applied Physics Laboratory at the University of Washington, the Department of Ocean Engineering at the Massachusetts Institute of Technology and by the Naval Research Laboratory in Washington, D.C. Acoustically controlled vehicles such as UARS (Undersea Arctic Research System) and SPURV (Self-Propelled Underwater Research Vehicle) are examples of this type.

Performance requirements of hovering and maneuvering at zero and low to medium speeds rule out the use of existing torpedoes as they are designed to have dynamical stability only at high speed, with course governed by movable control surfaces. They are incapable of maintaining course at low speeds and cannot hover. A second disadvantage is that the streamlined hull of the torpedo does not lend itself to the addition of appendages such as TV cameras, side-looking sonars and other inspection sensors.

The torpedo shaped vehicle generally has no capability for hovering or for sidewise motion, and thus as an inspection tool it is most valuable in high speed surveys over the sea floor.

The vehicle now in design at the Naval Ocean Systems Center, a complementary development to the UNH-MSEL system, is torpedo-like in form, with the addition of a thruster in the vertical axis. Although unequipped for sidewise motion, this vehicle does hover, and may be streamlined for effective pipeline and sea floor surveys.

The "open-frame" type of design, has been used extensively in the development of unmanned, tethered submersibles and is also being adapted to unmanned, untethered vehicles. "Open-frame" design features are exemplified in vehicles such as the U.S. Navy's family of CURV submersibles, Perry Oceanographic's RECON group and Saab-Scania's SAAB-SUB.

EAVE-East is an "open-frame" type of submersible and is ideally suited to meet the requirements of hovering and maneuvering at zero and low to medium speeds -- particularly in the absence of a requirement for high speed operation. Open-frame construction facilitates the positioning of thruster-units to obtain up to six degrees of freedom in maneuvering or to maintain position (hover) in any adverse current field. This type of construction also permits control and inspection sensors to be appended to the vehicle in the most advantageous locations for performing their functions. Sensor units can also be readily changed to provide the optimum sensor package for a given mission task.

2.A <u>EAVE - Platform Subsystem</u>

The general characteristics of the UNH Experimental Autonomous Vehicle (EAVE) are listed in Table 1. A picture of the vehicle in shown in Figure 1.

Basically, EAVE is an open frame vehicle of tubular aluminum construction. The frame acts as a mounting support for the following:

a) Two battery containers containing - sealed lead-acid cells, 2.5 AH size, providing power at 24, 12, and 6 volts. (Approximately 1 KWH per container)

b) Two electronics canisters, one containing the computer system, motor drivers and all other electronics, the second currently being empty.

c) Two large buoyancy tubes.

d) Six DC thruster motors.

e) Sonar sensor ring, consisting of 12 2ØØ-volt transducers pointing vertically downwards.

All containers are constructed of 6Ø61-T6 aluminum and seal with O-ring face seals. In its present configuration, the vehicle has a depth capability of 2ØØØ feet.

The thrusters consist of propellers driven by DC motors in an oil-filled housing with a diaphragm for pressure compensation. Each thruster motor is rated at Ø.25 horsepower and 17.Ø pounds of thrust.

In theory, the thruster arrangement provides 5 degrees of freedom (rolls not included). In practice, the placement of the batteries, low on the frame, and the buoyancy tubes high on the frame, results in a strong righting moment which severely limits the vehicle's ability to adjust its orientation in pitch.

The buoyancy tubes provide a net positive buoyancy of 5 pounds. The vehicle will, therefore, float to the surface if all thrust is turned off.

The open frame modular construction allows the addition of mission-specific sensors or equipment as required. This configuration does have the disadavantages of increased chance of entanglement in lines traveling about structure, and provides less protection for on-board items, if protective fanning is not provided.

2.B POWER SYSTEM

The power supply of EAVE has changed radically from its earlier configuration, for the current EAVE needed a dependable supply for up to 6 hours of operation while supplying power to the sonar and computer sections. As before, the main power load on the system consists of the 6 thrusters used to maneuver. Some other important considerations in the power supply design are:

1. High power density in a small package
2. Ease of maintenance
3. Quick and easy recharging systems

All of these considerations have been met with the selection of "D cell size" sealed lead acid batteries. These batteries provide 2 volts at 2.5 amp-hours. They can be arranged in various configurations to tailor a power supply for a particular application.

The EAVE system needs 3 separate, and not interconnected voltages, thereby reducing the noise transmitted by switching of DC motors. The voltages needed are 6V, for the computer system, 12V for the sonar and 24V for the thrusters. The computer, of course, needs only a minimal amount of power due to its CMOS architecture. The sonar system with its 12 volt supply, uses 12 batteries as 2 groups of 6 in series, tied in parallel. The thrusters cause the largest power drain, requiring 200 WH at normal working speeds, using perhaps 800 WH in a full days work.

The batteries are packaged, as previous mentioned, in tiers. The 24V supply requires 7 tiers, while the computer and sonar supplies are on a half tier. The tiers are divided into two groups when placed in the two battery pressure vessels. Each case has "O" ring seals and 4 bulkhead connections. Also, each case has a pressure release valve to vent gasses as they may occur.

The power leads from each tier are brought out to terminal strips for interconnection, recharging, and protection. Figure 2 details the strip connections for the 6, 12, and 3/7 of the 24 volt power source. Note that each tier of batteries is diode protected to prevent one tier from discharging another if a battery problem occurs.

Each tier is separated by 2 teflon sheets to prevent short circuits. Also, care has been taken to assure that there are no sources of sparks from switching transients within the battery case which might ignite any traces of hydrogen gas given off by the batteries. Altough gas is more likely during a recharging cycle, care is taken to open the entire battery case to vent any gas build up at that time.

A battery charging system is designed to recharge all sections simultaneously. The system uses a standard 24 volt car battery charger to charge the 24 volt section. The 6 and 12 volt batteries are recharged using two voltage regulators, one for 6 volts and one for 12, from the 24 volt line. Since there are fewer batteries in these sections, the power demand for recharging is small allowing the use of simple regulators.

The two identical battery cables employ standard 8 pin, high current male connectors. Their female counterparts are permanently mounted to the battery packs (See Figure 2). The recharging procedure is simple. The steps are:

1. Open both ends of each battery pack

2. Attach the battery recharging cables, one to each pack

3. Turn on the battery charger and adjust the rates and timer controls to charge the 24 volt sections. (The 6 and 12 volt regulators are preset and do not need any other adjustments).

The entire battery recharging process takes anywhere from 2 hours for a light refresh to 6 hours for a full, deep charge.

The last portion of the EAVE battery system is the battery tester, which tests if the batteries have received a full charge and are ready for operation. A load is placed across each battery section and the voltage drop monitored (thereby determining the current and power being delivered to the load). The 24 volt section must be able to deliver 20 amps for 20 seconds without a voltage drop greater than a 1/2 volt. The 2 volt sonar supply must deliver 3/4 amp for 30 seconds with no more than a 1/2 volt drop. The 6 volt section must supply 1/2 amp for 30 seconds without a drop of greater than 1/2 volt. These tests exceed actual requirements, for the CPU as well as the sonar draw much less than 1/2 amp. The thrusters require on the average, 5 amps with 4 thrusters operating to make an average load of 20 amps.

2.C EAVE MICROCOMPUTER SYSTEM

The EAVE microcomputer system contains a single bus structure that provides access to all components of the computer and, as such, permits use of a modular design concept which will accept additional function blocks with an absolute minimum of hardware change. This concept is necessary due to the evolutionary nature of a program where frequent system alterations and advances occur based on ever increasing experience with the system.

The microcomputer system may be divided into three functional areas: (See Figure 3).

1. The microprocessor with associated control and memory including the serial interface.

2. The motor control interface system.

3. The sonar interface system.

Area (1) contains the central processing logic as well as the control logic necessary to achieve physical responses. Areas (2) and (3) are concerned with the actual control of the vehicle in performing its task of locating and following, at a predetermined height, an exposed pipeline, as well as conducting the various support maneuvers.

2.C.1 EAVE COMPUTER SYSTEM (FIG. 3)

The computer as a system controller, is central to the vehicle's adaptive performance, possessing a capability to make appropriate decisions based on sensed inputs and stored logic patterns, and to give control commands to the vehicle's thrust-producing system for execution.

The microprocessor system may be subdivided into 3 sections. These are:

a) The IM6100 microprocessor

b) Volatile and non-volatile memory

c) Communications to the outside world

The Intersil IM6100 microprocessor was chosen among a wide range of candidate central processing units for two leading reasons. Firstly, it is of CMOS construction, and thus draws an extraordinarily small amount of DC power. Computer power demands, of course, are small when compared with thruster power requirements. It is anticipated however, that a mission could require long periods of quiescent sea floor operation where the continuing computer drain then might have an important impact on the power budget.

Secondly, the instruction set is identical to that of the long available PDP-8, and its use obviously saves programming training, permits use of proven diagnostics, and easy transfer of available software. The 6100, a 12-bit machine, is little less sophisticated than some modern competitors, but its competence is more than ample for the tasks associated with this vehicle, and perhaps for the vehicle that may follow it.

The IM6100 can directly address up to 4K words (12 bit words). Of this 4K, 1K is erasable-programmable read only memory (EPROM) and 3K words are random access memory (RAM). Extended memory addressing is readily available.

The EPROM is comprised of IM6553 CMOS PROMS. It is erased using ultraviolet light and is programmed using an IM6100 based PROM programming microcomputer*. The Programmable Read Only Memory (PROM) is used to store the software for the operating system. This non-volatile program, occupying octal locations 6000_8 through 7777_8, allows the user easy access to any address for inspection and/or modification (in RAM only). Some printing and timing programs are also located in PROM.

---

* For further information, see "The Engineering Design and Development of a Microcomputer Controlled Unmanned Free Swimming Vehicle", Final Report, 1978.

In any computer system some RAM is necessary and the IM6100 is no exception. In fact, the EAVE computer system is made up almost entirely of RAM, occupying locations 0000_8 through 5777_8. Due to the development nature of EAVE, the software is constantly being modified and altered. When the final version of the software is written, it will be burned into PROM with only a minimum amount of RAM left in the system. The Intersil 6518 1024x1 CMOS RAM chip is used to implement the RAM in hardware. Each RAM card has 12-6518 chips along with a few CMOS chips for addressing and latching of data (1024 x 12).

The microcomputer communicates with the user via a universal asynchronous receiver/transmitter (UART). It takes 8 bits of parallel data and transmits it in serial form. The speed at which the data is sent serially may be programmed either by switches located on the UART card or via a software routine. This is a necessity because different parts of the system can accept data at different speeds. For example, if communication is desired between two computer systems a very high data rate is acceptable. However, if an ASR33 Teletype were used, it can only receive and transmit at low speeds. Hence, many types of peripheral devices can be used with the vehicle computer system without hardware modifications to the UART.

The physical link between the UART and an external port is carried out via a modified 20 ma current loop. This provides a relatively noise insensitive communication link which can be accessed through long cables.

2.C.2 EAVE THRUSTER CONTROL

The motor control interface system determines the speed and direction of the DC motors of each of the six thrust-producing units. The motors are individually controlled by program instructions in the form of a four-bit motor control word which determines one of eight specific speeds, (2^3), and forward or reverse, (2^1). During the motor instruction sequence, the four-bit word is placed in a four-bit latch. This word remains in the latch until updated speed and forward or aft direction is called for by the control program. The motor speed is computer controlled by a pulse-width modulation system. The first three bits of the motor control word are used to set the average power available for the motor-drive circuitry. The motor direction is set by the fourth bit of the motor control word which determines the polarity of the applied drive voltage.

2.C.3 EAVE SONAR SYSTEM

The EAVE sonar system is responsible for two functions which affect vehicle performance. It provides data to the CPU that permits the vehicle to hover at preset depths throughout the mission. Secondly, the sonar data is analyzed to locate a pipeline lying on the ocean bottom, thus providing steering information.

The system consists of 12 equally spaced 200 KHz transducers mounted on a 5 foot diameter ring, 12 separate and complete sonar transmitter/receiver circuits and an interface card to communicate data to and from the CPU. (See Figure 4 & 5). The system is under direct control of the CPU.

On command from the CPU through the Sonar Interface (SI), timers are reset and a particular sonar circuit, one of twelve preselected by the CPU, is told to transmit an acoustic signal or "pinged". At that instant, the timers are started and the system waits a preset time for a return signal. When the signal returns, the timers are stopped and a number representing time, proportional to the distance the transducer is from the object, is transferred to the CPU for processing. If an acoustic return was not detected after the preset amount of time, an error signal is generated. The CPU will then take steps to rectify the situation (i.e. adjust the receiver gain, or raise/lower the vehicle). This scenario is repeated for any or all of the individual sonar circuits. A more detailed system breakdown follows.

The SI controls several functions of the sonar system. (See Figure 5A). These are:

a) Providing communications between the CPU and the sonar system via an Intersil IM6101 Parallel Interface Element (PIE).

b) Providing all necessary control signals to the timers and sonar circuits. This includes resetting of counters and timers, starting/stopping the counters, initiating a transmit pulse detecting a return from a sonar cirucit and generating error signals.

c) Timers to count two-way travel time and return the count to the CPU.

d) A delay clock that will start to count until a preset limit, (set by the CPU), is reached. Upon reaching this limit, it will signal the control logic that it has finished its count. The control logic will then generating an error signal signifying that a return has not been detected.

e) A hardware adjustable oscillator to clock the timers.

A single LC network is time shared to set up the transmit and receive frequencies. Also, a step-up transformer is added to gain more output power by appropriate impedance matching.*

The transducers operate at 200 KHz with a conical beam width approximately 7°. These units were chosen for two reasons. Firstly, the high operating frequency will reflect from almost all bottom features (i.e. the top layer of silt on the sea bottom, rocks, pipes, etc.), without penetration. Secondly, the LM1812 sonar chip operates as a high efficiency class C power amlifier, when used with this transducer.

The sonar system is, of course, driven by the computer. The software program "pings" each transducer 2 times and divides by 2 to form an average. It then averages the returns from all 12 transducers to obtain an "average of averages". It uses this number for two functions; - to compute the correction factor for adjustment of the vehicle's thrusters to permit hovering above the bottom, and to scan for any deviations in echo return length greater than a preset amount. The presence of a pipeline will cause reduced echo path lengths. This data enters an algorithm which corrects the vehicle course to align it over the pipe.

---

*For more information see: "A Single-Chip Monolithic Sonar System", T.M. Frederiksen and W.M. Howard, IEEE Journal of Solid State Circuits, Dec. 1974, Vol. SC-9, No. 6, pp.394-403.

The sonar system must cope with conditions which may inhibit successful pipeline following. Reflection of the acoustic signal from one transducer may be interpreted as the return signal of another transducer. Moreover, the vehicle may pass over a less-than-ideally flat bottom - a bottom strewn with rocks, debris, and other objects which may cause incorrect vehicle-bottom distance readings. Moreover, a rock is not intrinsically different as an echoer than an exposed pipeline. Consequently, it is important to the decision-making process that a criterion capable of differentiating between correct and false readings be employed. A digital gain control feature of the sonar system along with the microcomputer system and associated software accomplishes this differentiation. If a reading is determined to be false, the gain of that transceiver may be altered so as to eliminte the false returns. This is done by latching a new, four-bit, gain-control word into the D/A converter which, in turn, alters the voltage applied to a set of FET's used as voltage variable resistors (VVR). It is this resistance which sets the gain of a particular transceiver circuit. The four-bit word thus allow sixteen, (2^4), levels of gain.

Figure 1

UNH VEHICLE

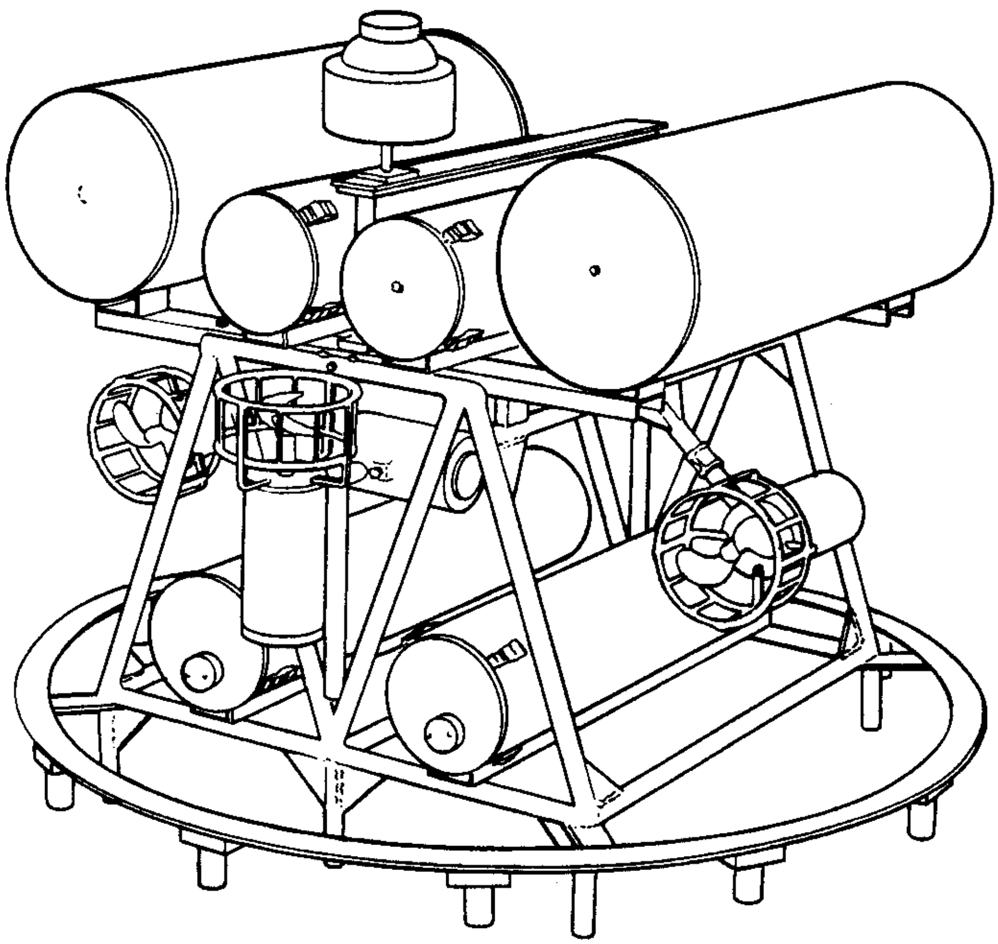


Table 1

OPERATING CHARACTERISTICS

PHYSICAL CHARACTERISTICS

PLATFORM	
OVERALL DIMENSIONS:	
LENGTH	4'
BREADTH	4'
HEIGHT	3'6"
DISPLACEMENT SUBMERGED	691.5 LBS.
WEIGHT	687.4 LBS.
POSITIVE BUOYANCE	4.1 LBS.
PAYLOAD	SEE NOTE
STATICAL STABILITY (BG)	0.40 FT. (POS)
SPEED (MAX - NO CURRENT)	2.0 KNOTS
POWER (AT MAX SPEED)	0.50 H.P.
MANEUVERING (THRUST/MOMENT)	
X - AXIS	
SURGE	34.0 LBS.
Y - AXIS	
SWAY	34.0 LBS.
PITCH	136.0 FT. LBS.
Z - AXIS	
HEAVE	34.0 LBS.
YAW	85.0 FT. LBS.
POWER SYSTEM	
LEAD/ACID	
GEL CELLS	+12 VDC
"D" SIZE	+24 VDC
2.5 AMP HOUR	+ 6 VDC
TOTAL ENERGY	APPROX. 2KWHR.

NOTE: Payload = "Mission Equipment" for pipeline survey & monitoring. Space frame can easily carry buoyancy packages supporting 100 lbs. payload when on board.

OPERATING CHARACTERISTICS

SPEED (MAX SUSTAINED-STILL WATER)	2.0 KNOTS
MANEUVERABILITY	5 DEGREE FREEDOM
HOVERING CAPABILITY	YES
REVERSE CAPABILITY	1.5 KNOTS
MISSION DURATION	4-10 HOURS
DEPTH (MAX OPERATING)	2000 FEET
SENSORS/COMPUTER	
SENSOR SUIT	
CONTROL SENSORS	TIME, ALTITUDE HEADING DEPTH INTERNAL PARAMETERS FAILSAFE SYSTEM
DATA SENSORS	ACOUSTIC PIPE FOLLOWING
COMPUTER	
CPU	3-INTERSIL 6100 MICROCOMPUTER SYSTEMS A. CONTROL B. COMMUNICATION C. NAVIGATION
COMMUNICATIONS	DIRECT TETHER ACOUSTIC LINK
NAVIGATION	PREPROGRAMMED INTERNAL DEPTH/HEADING/TIME EXTERNAL ACOUSTIC MONITORING

Figure 2

BATTERY PACKS

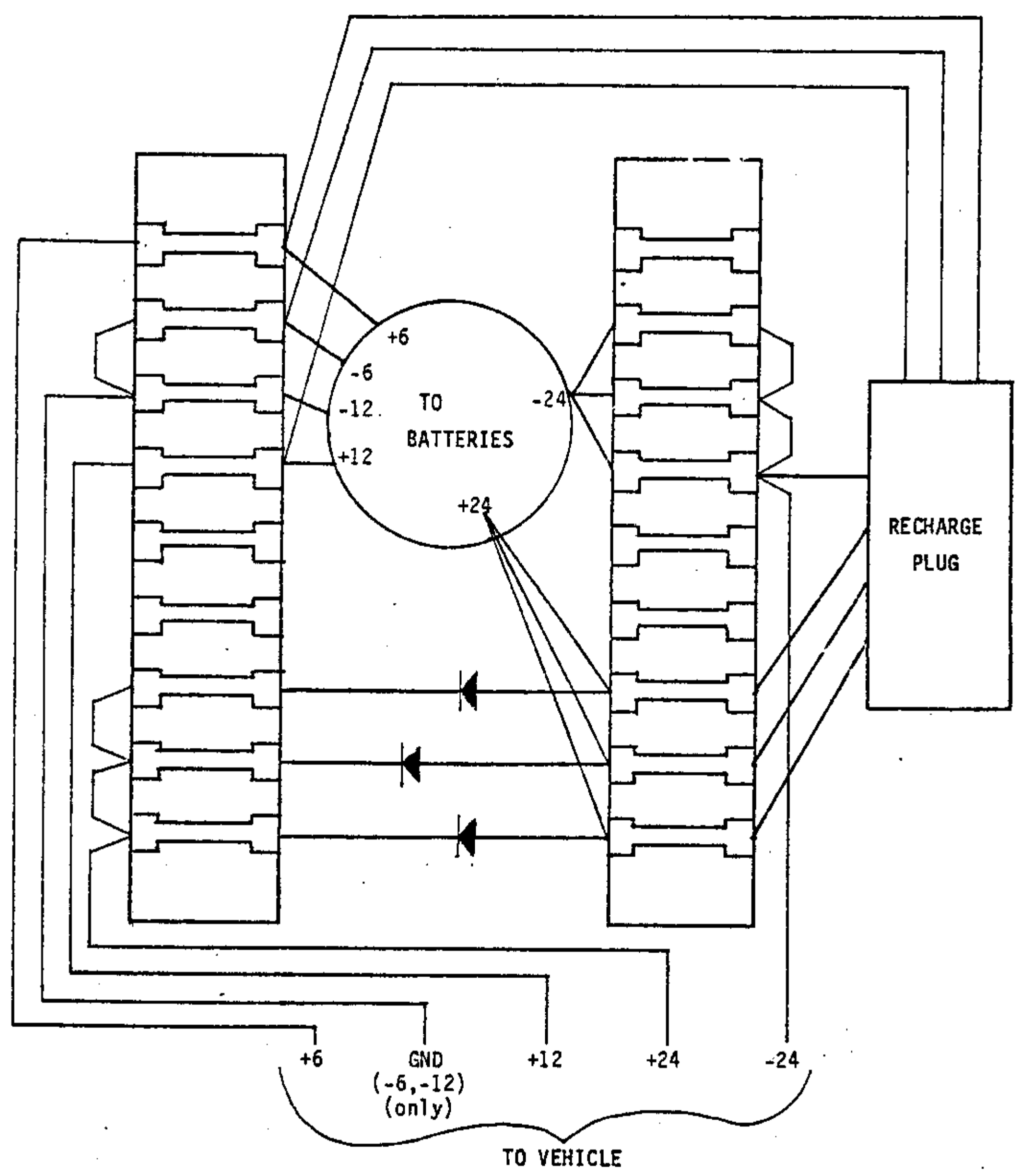


Figure 3

MICROCOMPUTER SYSTEM BLOCK DIAGRAM

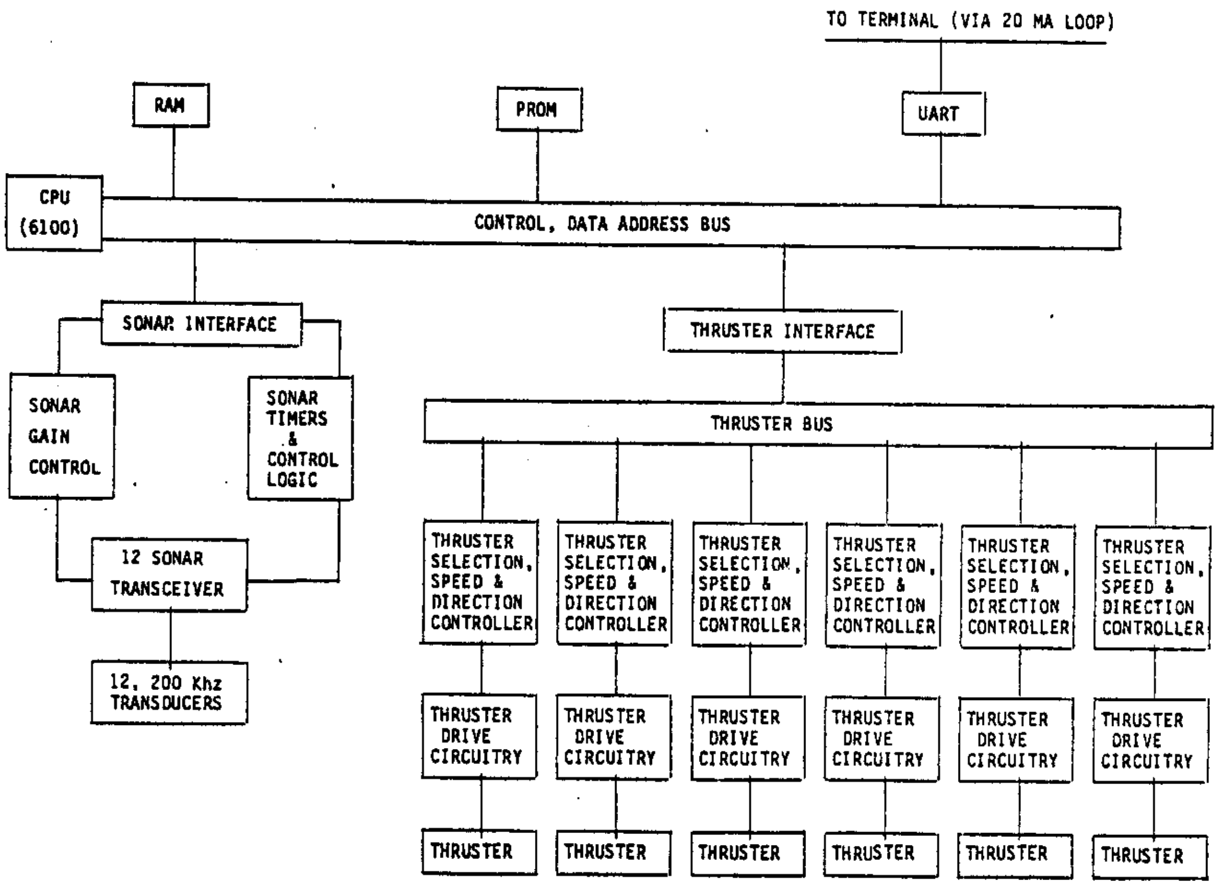


Figure 4

COMPLETE SONAR SYSTEM

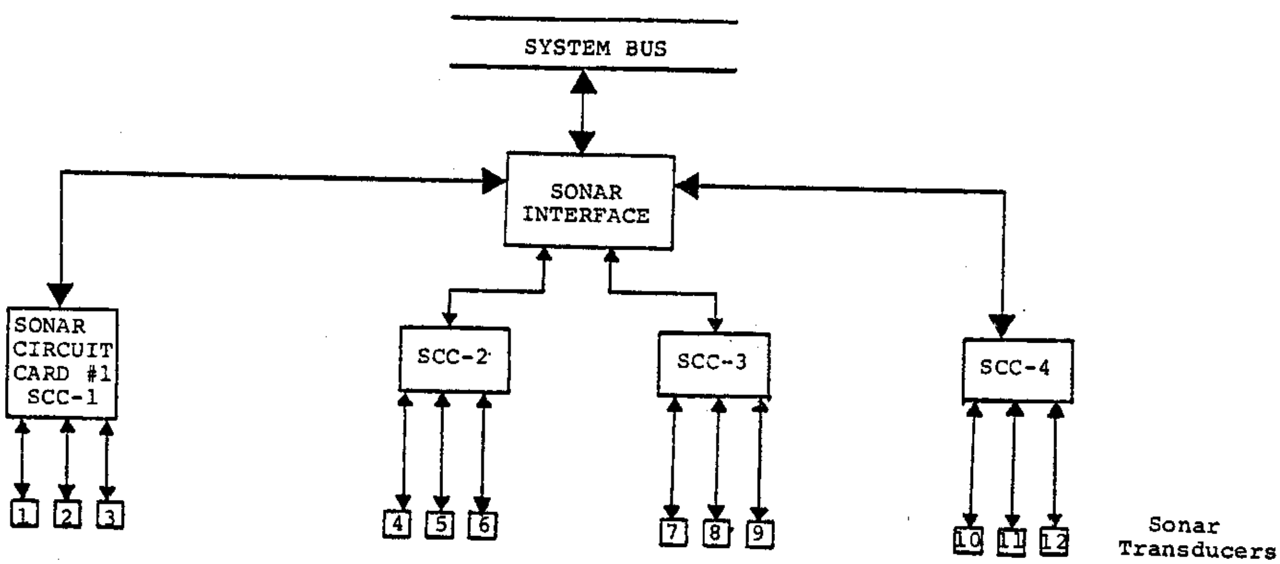


Figure 5

SONAR INTERFACE

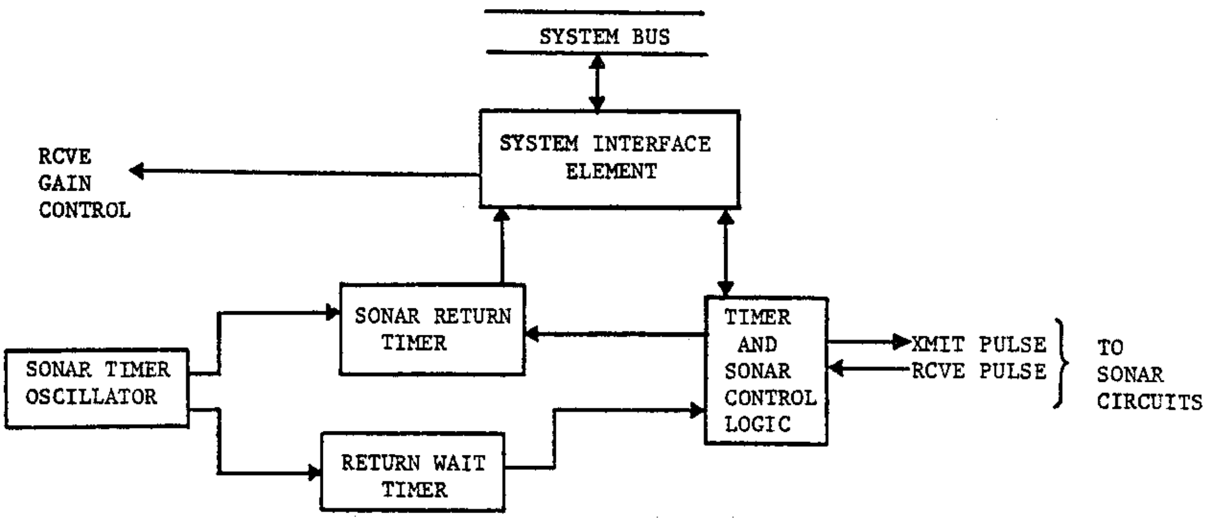
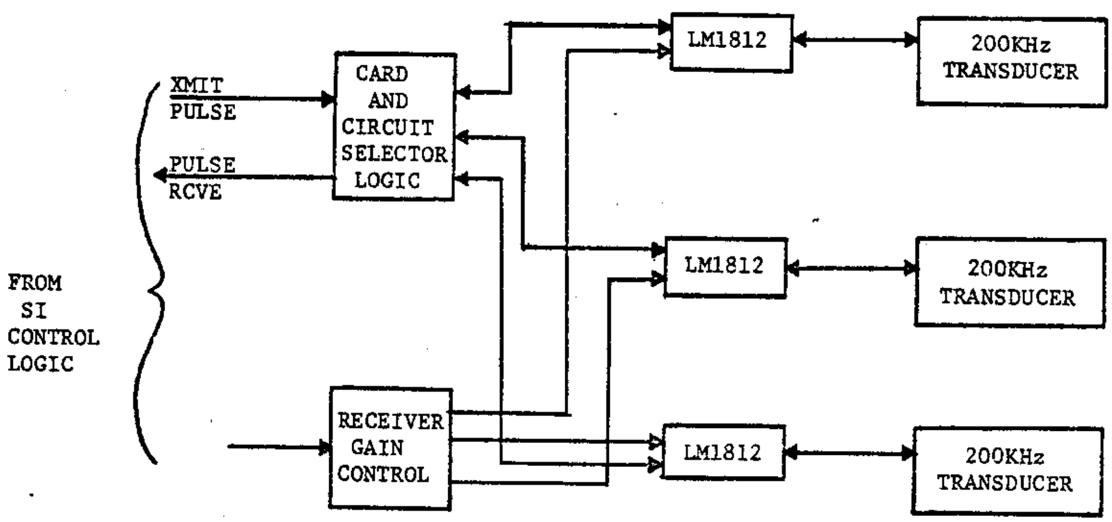


Figure 5A

ONE OF FOUR SONAR CIRCUIT CARDS



3. SOFTWARE SYSTEM

The software library created for the Experimental Autonomous Vehicle, developed at the Marine Systems Engineering Laboratory, consists of four major groups of programs written in machine and assembly language using the Intersil IM6100 instruction set. The UNH Dec-10 main frame computer was used for the creation, editing, and assembly of the software, making particular use of the PAL 10 assembler. With the acquisition of the DEC station 78 development system, virtually all of the above software tasks including storage are now carried out within the Laboratory itself, freeing the Lab from the time constraints of the main frame computer.

In-house program modules written in Fortran-10 are used to convert each assembled subroutine into an octalized file form that is loaded by the IM6100 CPU into the 4K of available RAM. A set of MSEL created monitor modules are contained in EPROM and used for CPU control, software loading and debugging and I/O cperations. These software modules serve either to control the vehicle in the autonomous mode or to allow an operator to control the vehicle by means of a terminal connected to the on-board computer by a tether, or subsequently by acoustic command link. The four major program groups are:

1. Thruster Interactive Diagnostic Routines
2. Pipefollower Programs
3. Maneuvering Programs
4. Trackline Printout

These are described in detail below. Their flow diagrams are included in Appendix E. Copies of machine language printouts and written descriptions of the flow diagrams are not included, for space reasons, although the sub-routines are named in the descriptions below. Correspondence is invited on the details of these programs.

3.A Thruster Interactive Diagnostic Routines

A. General

The Thruster Test Programs allow the operator to test the operation of the six thrusters and their seven speeds forward and seven speeds reverse.

The Thruster Test Programs are:

Thrus 2, A, S

B. <u>Operation</u>

Upon execution of the programs, the routine will wait for the operator to hit any TTY key and as soon as this is done, it will print the introductory instructions. The operator may then choose one of three command instructions as follows:

Type "A" - The Program will run through an automatic test sequence starting with thruster number one and going through all seven speeds from maximum forward to maximum reverse for all six thrusters. If any TTY key is hit during the sequence, the thruster will shut off and control will jump to ODT.*

Upon completion of the sequence for each thruster, the program will ask the operator if he wishes to test full power reverse and wait five seconds for any TTY key to be hit. If no key is hit, the program will continue testing the next thruster.

---

*ODT - Octal Debugging Technique.

Type "S"- The Program will ask the operator which individual thruster he wishes to test and at what speed. The user may then increment the speed by typing "I", decrement the speed by typing "D", or enter a new speed. If the operator types "S", the thruster being tested will shut down and control will return to ODT.

Type "X" - The Program will send control to ODT.

3.B <u>Pipefollower Programs</u>

3.B.1 <u>General</u>

The pipefollower programs will allow the vehicle to search for, position itself over, and follow a submerged pipeline. There are also provisions for manual control through a tether and a search routine in the event that the pipeline is lost. The pipefollower programs are:

VAR, KEYHIT, FOLOWR, XDUCER, ALTUDE, WHERE, STEER, AVERAG, SLIDE, LRTURN, UPDWN, FORBAC, MES, PRINT, DIRECT, REG, COMMON.

3.B.2 Operation

The software library created specifically for the pipefollowing mission may be divided conceptually into four major groups of modular subroutines according to their prime responsibilities. The four module groups are:

1. Vehicle control-master
2. Data acquisition
3. Data processing
4. Vehicle control-autonomous

The order presented here and in which these modules will be discussed follows the order of execution during one complete operating cycle of the pipefollowing mission. The operating cycle time using the current software library is approximately one second.

3.B.2a Vehicle Control - Master

The vehicle control-master module consists of the subroutine KEYHIT. This subroutine allows operator control over all other software modules in the EAVE software library.

The following operations may be performed as a result of engaging this primary routine:

A. The monitor may be called to edit, debug, and run all software contained in accessable memory.

B. The vehicle may be maneuvered directly by the operator by calling selected subroutines to alter the status of any or all thruster motors.

When the vehicle is connected by the tether to a teletype, the operator may navigate the vehicle at any time by typing in the following commands:

"F" - Go Forward	"L" - Turn Left
"B" - Go Back	"R" - Turn Right
"U" - Go Up	"X" - Slide Left
"D" - Go Down	"Y" - Slide Right
"O" - Go To "ODT"	"S" - Stop All Thrusters

This is followed by the speed number desired.

"O" - Shut down the appropriate thruster.

Example: "FO" will shut down the forward thrusters.

"1" - Set the appropriate thrusters to slow speed.

"2" - Set the appropriate thrusters to medium speed.

"3" - Set the appropriate thrusters to fast speed.

These selectable subroutines are referred to as the maneuvering routines and are called "FORBAC", "UPDWN", "LRTURN", AND "SLIDE". They set the actual motor speeds and directions according to the parameters passed to them by the control modules.

C. All pre-written subroutines that control autonomous activity may be called and run either with the cable-tether connected, or with the tether removed.

3.B.2b Data Acquisition

This subroutine called "XDUCER" controls the operation of and the collection of data from the 12 sonar transducers.

Each sonar transducer is pinged twice and a running total of the two-way times is stored in memory. If there occur four consecutive bad readings at a certain gain, the gain is raised to the next higher gain, up to fifteen gain levels. If all fifteen gains give bad data, the vehicle will give a timed downward thrust and will repeat the sequence until good data is obtained.

3.B.2c Data Processing

These subroutines process the data received from the transducers in two distinct ways. They are "AVERAG" and "WHERE".

A. First, the 2-way travel times of each transducer are averaged then, these averages are in turn averaged to form a base value which may be used for comparison purposes and as a value for the actual altitude of the vehicle above the bottom.

B. Second, the 2-way travel time average for each transducer is compared with the base average. the transducer average is found to be less than the base average by a predetermined constant, a bit is set in a memory location to note the presence of an object such as a pipeline under that particular transducer. Another subroutine then looks at this memory location and sets one or more bits in another memory location in groups of three transducers each. Each of the four groups of three transducers is termed a quad and is used to represent the presence of an object in any one or more quadrants of area as is shown in Figure 6.

The memory location "quad" is then examined to determine the number of quads the pipeline is seen to be under. If more than 2 quads are set, this is determined to be an error condition due to inaccurate sonar returns, reflections, etc. When this occurs, the last command that was executed as a result of the acquisition of good data is continued and program flow returns to start a new operational cycle.

3.B.2d Vehicle Control - Autonomous

The autonomous control modules are the heart of the pipefollower mission software. They include the subroutines FOLOWR, ALTUDE, and STEER. They make decisions based on the data processed above and set thruster motor speeds and directions to accomplish two distinct operational goals.

A. First the subroutine "ALTUDE" is called. Here the base value figured calculated above is taken to be the actual vehicle height above the bottom. This figure is compared to a constant placed in memory which represents the desired altitude above the bottom. These two figures are compared and thruster motor speeds and directions are set to bring the two figures into allignment, thus maintaining a constant altitude for the pipefollower mission.

B. Second, the subroutine "STEER" is called. This subroutine examines the memory location "QUAD" to determine the location of the pipe beneath the vehicle and sets thruster motor speeds and to center the pipe under the fore and aft transducers and propel the vehicle in a forward direction thus following the pipe.

The subroutine "FOLOWR" controls the program flow of autonomous activity, using approximately one second for one complete mission cycle.

Within "FOLOWR" are two programmed delay loops that enhance the flexibility of the mission software.

The first creates a timed delay that allows the tether to be removed and the vehicle to be lowered into the water before the autonomous software is engaged.

The second is a mission abort timer that allows the setting of a maximum number of software cycles for a particular mission. Upon execution of the desired mission time, all thruster motors are shut off and a timed up thrust given. Program flow then returns to the master control module while waiting for the tether to be reattached and manual control returned.

When the pipefollower software is first engaged, the vehicle will first secure the desired operational altitude before engaging any other subroutines. With this accomplished, the forward thrusters are engaged at slow speed and maintained at this speed as long as no pipe is seen. With the setting of the first bit in "Quad" the pipe is considered to be found and a flag is set. The full pipefollower cycle is then performed continuously. If at any time the pipe is lost, that is no bits are set in "Quad", the found 'flag' is reset and a 'lost routine' engaged until the pipe is re-acquired, hereupon the mission proceeds as before. The "Lost Routine" performs a slowly expanding search spiral. A delay of five cycles is built into the start of lost routine to ensure that the pipe is actually lost and that the condition is not due to bad data being received from the sonar transducers. The vehicle will perform the lost valid command for five cycles before engaging the lost routine.

3.C Maneuvering programs

(See Figure 7 - Thruster Orientation)

3.C.1 General

The maneuvering programs allow the operator to maneuver the vehicle by typing various commands on a teletype. Communication with the on-board computer is accomplished by means of a multi-channel cable tether, with acoustic telemetry under development.

The programs may be implemented in two distinct ways as determined by the specific programs loaded into computer memory. In the first, the vehicle will seek and maintain a predetermined altitude above the bottom while responding to operator maneuvering commands. This is referred to as the "auto-altitude" mode.

The specific programs used for maneuvering the vehicle with the auto-altitude mode are:

TMAIN, AUTO, SLIDE, LRTURN, UPDWN, FORBAC, DIRECT,
REG, COMMON, PRINT, MES

In the second case, the vehicle will respond solely to operator maneuvering commands with no altitude seeking provisions.

The specific programs used for maneuvering the vehicle without the auto-altitude mode are:

OLDTM, SLIDE, LRTURN, UPDWN, FORBAC, DIRECT, REG, COMMON, PRINT, MES

3.C.2 Operation

Upon execution of the programs with the auto-altitude mode included, a single, selected transducer is pinged eight times and a running total of the two-way times stored in computer memory. If there occur four consecutive bad readings at a certain gain, the gain is raised to the next higher gain, up to fifteen gains. If no good data are obtaind at the highest gain, the vehicle will execute a five second downward thrust to decrease the altitude of the vehicle above the bottom, and will repeat the sequence starting at the lowest gain until good data is obtained.

When eight good data sets are obtained, the average is calculated and compared to a predetermined two-way time for the desired operational altitude. This comparison will generate the proper thruster response to correct the vehicle's altitude.

At any time, the operator may maneuver the vehicle by typing the appropriate commands on the teletype as follows:

"F" - go forward	"L" - turn left
"B" - go back	"R" - turn right
"U" - go up	"X" - slide left
"D" - go down	"Y" - slide right
"O" - go to "ODT"	"X" - stop all thrusters

This is followed by the speed number desired.

"Ø" - shut down the appropriate thruster

Example: "FØ" will shut down the forward thrusters.

"1" - set the appropriate thrusters to slow speed.

"2" - set the appropriate thrusters to medium speed.

"3" - set the appropriate thrusters to fast speed.

An "L" or "R" can only be followed by a "1", "2", or "3", but not "Ø". If any key is struck while the vehicle is makine a left or right turn, all thrusters are shut off and control goes to "ODT". This serves as an emergency stopping routine.

The system will execute the operator maneuvering commands until the operator hits any teletype key releasing it to return to the auto-altitude mode.

Upon execution of the programs without the auto-altitude mode, the vehicle will respond to the operator commands only and will make no independent altitude corrections.

3.D. Trackline Printout

3.D.1 General

The trackline printout programs will allow the vehicle to dive to a specified altitude above the bottom and go forward while taking sonar readings. The readings are stored in computer memory. When a specified number of readings have been taken, the vehicle will surface and use the data obtained to print a profile of the bottom.

The trackline printout programs are:

TMAIN, SONAR, PIN62, PICTUR, DIRECT, REG, COMMON, PRINT, SLIDE, LRTURN, UPDWN, FORBAC

3.D.2 Operation

When the desired altitude is obtained, as described earlier, the programs will cause the vehicle to power at a slow speed while pinging the transducer. 512 samples of data are then taken in 64 groups of 8. Each data set of 8 readings is inserted into the low end of a storage array after older data sets are pushed upward. After each data set is obtained, the program will make an altitude correction. When all 64 data sets are obtained, the vehicle will surface, and at the same time process the data into printable form. This is done by forming an average for each data set of 8 readings. This average is subtracted from each of the 8 readings in its data set. The result of each of the subtractions is then divided by 7 and placed back in the original location in the storage array. When the array has been fully processed, the program will print the array on the teletype terminal. Variations in elevation above the bottom, such as a pipeline, may then be easily observed.

If a tether is used, the processing and printout will begin as soon as the storage array is full, indicating that all data has been received. In addition, operator maneuvering commands may be used at any time to navigate the vehicle. These are described in sections dealing with the maneuvering programs.

When no tether is used, the processing and printing routines will wait for an operator command. This allows time for the tether to be connected.

Figure 6

QUADRANT LAYOUT FOR THE 12 SONAR TRANSDUCERS

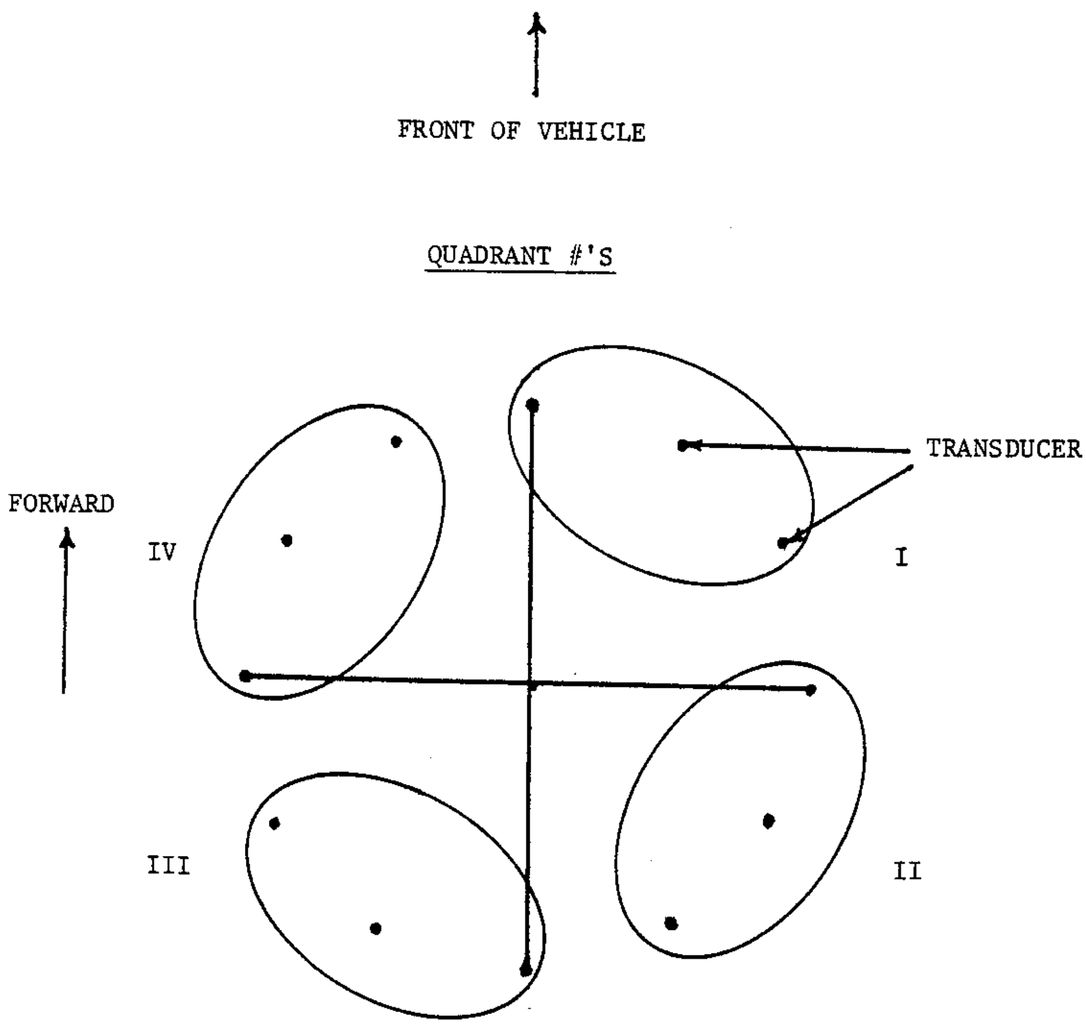
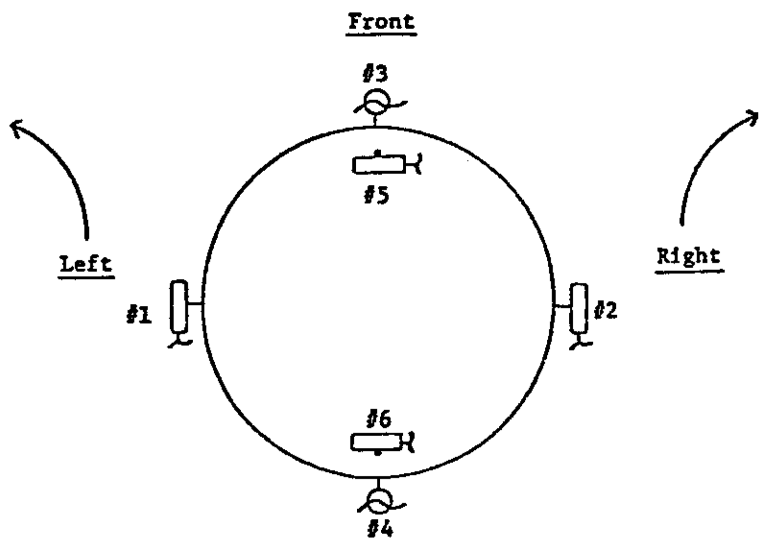


Figure 7

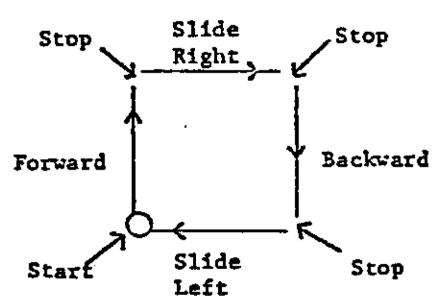
THRUSTSER ORIENTATION



Thrusters #1 & #2 are used to move the vehicle forward or backward, and also to turn the vehicle left & right.

Thrusters #3 & #4 are used to move the vehicle up or down.

Thrusters #5 & #6 are used to slide the vehicle to the left or right.



SQUARE

4. NAVIGATION

4.A. Overview

Under Contract No. N66ØØ1-8Ø-C-ØØ5Ø, a pair of tasks were authorized relating to navigation system design and evaluation. This contract was in addition to, but totally supportive of Contract No. N66ØØ1-79-C-ØØ55.

A test range was to be established in Lake Winnipesaukee involving three hydrophone stations. Systems were to be tested and accuracy determined.

The homing/location system was to be developed and tested.

The consequences of this study are inclued in Section 4 o~ this report.

4.B. High Resolution Navigation System

4.B.1 Introduction

A knowledge of current position and its rates of change is a critical input to an autonomous vehicle. A frame of reference, ideally in geodetic terms, must link the vehicle, its point of origin, and its destination if the mission is to be accomplished and if the well-being of the vehicle is to be maintained. No other element of the vehicle system, however, is so intimately linked to the mission itself, or to its environment. Ideally we would want an all encompassing navigation system that serves equally well for deep water transects, for long range pipeline inspections as well, and for precision transits through the legs of an offshore tower. Each task, however, places widely differing demands on navigation, and there are few commonalities in hardware or specifications to be seen. Systems, specific to a class of mission, therefore, must be employed.

Since there is no universal system to be developed, a navigation system was selected at MSEL that best serves to measure the performance of this vehicle while under development, and a system whose components may serve as models for the system used in the future autonomous structural inspection mission.

This system requires a set of hydrophones emplanted on the fringes of the work area, and creates a grid, in terms of time of acoustic transmission from the vehicle that may be related quite accurately to geodetic position.

4.C <u>System Configuration</u>

Figure 8 illustrates the EAVE Test Range in Lake Winnipesaukee, NH, where the vehicle attempts to follow a pipeline around a convolved course. Three hydrophones, seen extended in the water column, serve as points of reference for a navigation grid. A measurement of the acoustic path length between the vehicle and each hydrophone, located at a precisely known position, will serve for computing the coordinat}s of the vehicle. Several modes of positioning are available to us, each with its advantage.

a) The vehicle may transmit pulses which are received at the set of passive hydrophones at times consistent with distance, and then transmitted by cable to a data processing center for computation and display.

b) The base hydrophones may transmit simultaneously, causing the passive vehicle to receive three appropriately coded pulses from which its on-board computer can compute its position dynamically.

There are variations on these systems. If transponders are employed, for example, the time lapse will permit calculation of range. If, however, a precise time reference is available at both the vehicle and the shore station, and if pulses are initiated at times known exactly, then range may be calculated from received signals alone.

The availability of simply the three pulses, without a reference, permits measurement of two time differences, which in turn allows calculation of position by Hyperbolic intersections. Knowledge of only two ranges permits calculation of position by Circular intersection, with the third measurement removing the ambiguity. Circular intersection navigation is superior, largely for practical reasons.

The EAVE program team chose as their approach, the system we call Passive Navigation (PNAV) employing the active vehicle transmitter and a set of three passive receivers. Synchronous clocks cause the vehicle to emit pulses at times known precisely to the receiver, permitting range calculations to be made at the shore station. Circular navigation is employed. The chosen system has the advantages of:

a) requiring only a simple transmitter on the vehicle, while permitting accurate location of its position by the shore station for monitoring of vehicle performance on the Test Range.

b) permitting a study of the accuracy and performance limitations of acoustic navigation in a shallow water environment.

c) permitting, in due course, a reversal of the geometry, whereby the vehicle receives a triplet of pulses from active hydrophones. It then may compute with its on-board computer its own position, and thus begin to truly become autonomous.

d) allowing the PNAV system to be extended to three dimensions to serve the critical navigation studies associated with the Structural Inspection mission.

e) achieving the desired 1 meter accuracy at a range of up to 300 meters in shallow water.

f) relatively simple software algorithms.

g) measurement of large numbers for range, rather than the measurement of differences in large numbers as required in Hyperbolic Navigation.

4.D System Boundaries

Acoustic measurements are plagued by many uncertainties, and acoustic navigation systems are no exception.

- Errors in position are propagated as a function of error in the location of the navigation buoys. Since precision in surveying of underwater stations is sometimes difficult to obtain, these errors may be significant.

- Errors in estimates of the arrival of the acoustic signals may be generated by multipath or by water-borne noise. Short multipath delay times are believed to be a major hazzard of the planned structural navigation system which eventually will grow from our learnings from this system.

- Acoustic shadowing by intervening structures can cause difficulty.

- Reverberation from the water mass may disturb estimates of the time of arrival of the acoustic pulses.

4.E System Parameters

In Figure 9 may be seen the Block Diagram of the Passive Navigation system.

The system parameters have been chosen to minimize error, and to reduce complexity, while evolving a first level navigation system prototype. Among the provisions made are:

A precision clock on the vehicle initiates a pulse at a time that is known to the shore based computer system, which has an identical, and synchronized clock. There will be drift in the clocks, and in due course they will produce error. The clocks in the EAVE tests were not temperature stabilized, due to the relatively short test runs between synchronization opportunities. It is noted that in the event of drift of the clocks, an error is added equally to each of the three measured ranges. Since the circular navigation system seeks the intersection of 3 range circles, an error will either result in a fix which is vlaid in the horizontal plane, with ambiguities in the depth plane, or if the drifts are subtractive, there will be no unique solution indicated. A triangular area of error will be defined by the 3 intersections of each range circle with the other two. In the first case, the pressure sensor resolves the ambiguity, in the second case, a software algorithm will be employed to approximate the proper point of intersection.

The hydrophones were placed in a configuration that included a right angle, as seen in Figure 10. This simplified the computations to be made by the shore based computer, and saved programming time - a not inconsequential matter. The base lengths for this configuration are fixed numerical parameters, while the three ranges (vehicle to hydrophone) are calculated by counters that are gated by the time of transmission. The height of a triangle, in this case the Y coordinate of vehicle position, can be determined by the following algorithm. (See Figure 11).

$$S = 1/2\ (A+B+C) \tag{1}$$

$$K = (S/(S-A)(S-B)(S-C))^{1/2} \tag{2}$$

$$h = 2K/C \tag{3}$$

Since precise determination of the required right angle is quite difficult in the water, it is possible that error does exist in the hydrophone configuration that would affect the estimates of vehicle position. This error is examined in Appendix F, and is found to be minimal.

4.F PNAV System Hardware

The PNAV system has two hardware centers. The main computational hardware is located at the shore station. Other hardware is used remotely at the vehicle station. The shore station hardware consists of a 6100 microprocessor, 3K of PROM, 1K of RAM, a UART and a custom interface.

The interface performs the timing and control functions for the computer. Timing is critical. A very stable 10 MHZ oscillator provides synchronization information for the remote station and the vehicle station. When the vehicle transmits a burst, the control circuit at the shore station clears a set of 3 4 KHZ counters. The counters are stopped by a negative edge pulse which is generated by a set of tone decoders when they receive the acoustic burst. Since the speed of sound in fresh water at 15°C is essentially 4900 ft/sec., and the count frequency is 4900 counts/sec., one bit, or count corresponds to 1 foot travelled by the burst. The truncation error due to roundoff then is $\pm 1/2$ bit or $\pm 1/2$ ft. or $\pm .15$ M. All clock signals are derived from a DALE 10 MHZ ocillator with a drift figure of .1 ppm over the temperature range of use.

Three acoustic receivers, one for each transponder are used to generate the stop pulses for the counters. The acoustic amplifiers have 60 DB of gain at 27 KHZ, the burst frequency, and have a 3 DB bandwidth of 2 KHZ. The amplifiers have AGC and provides the proper input signal for the 567 tone decoders. The tone decooers detect with ±.185 Msec ambiguity which corresponds to ±.89 ft or ±.27M.

The vehicle hardware consists of an identical crystal clock, timing hardware, a 27 KHZ gated oscillator and a power amplifier. The timing hardware, which runs synchronous with the shore station, gates the oscillator to produce a 2 msec burst of 27 KHZ output once every second. The burst drives a power amplifier which supplies 20 watts to the transducer. The drift of the oscillators at .1 ppm produces a drift of 1.7 ft/hr. or .54 M/hr.

The significant contributors to system errors as described above produce an RMS error of .62 M at 1 hour of operation.

4.G PNAV System Field Tests

The PNAV system was tested at the UNH - Lake Winnipesaukee test facility in September and October of 1979. The test goals were to verify the operation of the system and to measure the system performance.

The system performance was verified by causing the transmitter to swim around the pipeline route on the bottom of the lake and then reproducing its path on the screen of the video display. This test was performed several times and one such test was recorded on film.

Prior to, and after, completion of each test, the remote hardware was retrieved and the drift of the transmitter with respect to the shore station was measured. The drift was recorded at .3 msec at 1 hour. This was within design specification. It should be noted that this was also at the limit of the precision of calibration equipment used to adjust the hardware before the test.

The detect error of the phase locked loop was measured at a maximum of ±.18 msec. Typical error was much less than this and was on the order of ±.08 msec.

Using a drift of .3 msec and a detect error of .18 msec and adding in the 1/2 bit counting ambiguity gives an RMS distance error of .55 M. at 1 hour.

4.H Software Design

The software for the Passive Navigation System, written in the machine language of the Intersil IM6100, and compatible with the DEC PDP-8 serves two modes of operation; initialization and tracking. The display, made by Intertec, is not an ideal graphics terminal, although it does have some character graphics capabilty.

4.I.1 Initialization

The main purpose of the initialization routine is to synchronize the two separate 10 MHz clocks required for operation. The software causes a pulse to be sent through a PIE (peripheral interface element) which starts both the on-shore clock and the vehicle clock. The vehicle clock is hard-wired to the computer interface temporarily for this initialization. Once initialized, the clocks are separated and are left free running. (See Figure 12).

A secondary purpose of this routine is to set the parameters given for the physical configuration (eg. base lengths, scale factors, etc.) in page zero RAM. (Page zero consists of the first 200 octal memory locations of the computer). Since these parameters are constant for a given test site, they are set by the software for time and convenience. Each of the parameters, however, is easily changed in RAM once initialization is finished. This routine also initialized counters and sets the desired counter frequency. Due to the speed of sound in water, this frequency is set at 4.8 KHz so that one count corresponds to one foot traveled by the "ping". Thus, no scaling is required for the counter data once received.

4.I.2 <u>Display Design and Tracking System</u>

The tracking routine is the main program for the PNAV system. The routine calls various subroutines to get the data and to calculate the coordinates, and then plots the coordinates on the video display.

Since the full screen of the display represents the entire working area for the vehicle, exceptional resolution is not available. However, the system has the capability of expanding any quadrant of the working area to the full screen of the video display. This is done by processing the coordinates through a series of expansion variables, given the proper user command. Thus, the software starts by resetting these variables to the normal full field values.

The routine then begins the tracking procedure by calling the subroutine which receives the data from the counters. If an error occurs during this procedure, the link is set and then checked upon returning to the main program. If the link is found to be set, the program will repeat this process and look for more data.

If the link is not set, the routine continues and calls the subroutine which calculates the coordinates for the given set of data and scales it down so that it may be used to plot the data on the video display. Once these coordinates are formed, they are checked against the last set of coordinates to be plotted. If these coordinates are the same, the new data is ignored and the process is repeated. If a change has occured, indicating vehicle movement, then the coordinates are processed through the screen expansion routine and are either adjusted or left the same, depending on the mode of operation. These results are tested to see if the resultant coordinates are representable on the field of the display. If not, then the vehicle is out of bounds and a bell is rung, indicating the error. (eg. if quadrant 4 is expanded when the vehicle is in quadrant 2). The program then returns to the beginning. If the coordinates are within bounds, a reverse-video, half-intensity flashing X is printed in the appropriate location on the screen using cursor control and a normal, full-intensity dot (period) is placed in the location which last held a flashing X. The Intertube is limited at this point due to the fact that the screen can only print 24 rows and 80 columns of characters. Thus, data must be scaled to fit the screen. Once these points have been plotted on the screen, the process is repeated and new data is obtained.

The routine which receives the data from the counters handles the timing and counter interfaces and checks for errors in data. The first operation in the routine is to insure that the counters are all stopped and to reset the counters to zero before the start pulse is received. The start pulse signals the counters to start counting and occurs at the same time that the transmitter "pings". Once the counters are reset, the routine waits for the start pulse. After this pulse is detected, the routine waits a reasonable amount of time in which all stop pulses (or received signals) should have occurred. This time is basically a function of the working area of the system and is usually slightly larger than the largest possible travel time between the vehicle and any receiver within this area. Thus, if a pulse is to be received, it will be received within this time period. When this wait loop is finished, each of the three counters are checked to see if they have stopped counting (received stop pulses). If any counter has not stopped, an error counter is incremented and the link is set so that the data will be disregarded and the next set of data will be obtained. Each data point is also checked to insure that no length (L1, L2, L3) has changed by an unreasonable amount, given the rate of travel of the vehicle. If it has, this data is also discarded. This procedure is repeated for four sets of data. Then, to minimize any further error that might be present, the resultant for data sets are averaged to obtain one data set. This data is then used for computation and plotting.

The routine responsible for determination of the coordinates to be used for plotting works primarily on the basis of a floating-point math package based on Digital Equipment Corporation's PDP-8 Floating-Point System. This package operates on a three-word data field (36 bits) and performs addition, subtraction, multiplication, division, and square roots with 12 exponent bits (including sign of exponent bit), 1 sign of mantissa bit and 23 mantissa bits. Additional non-arithmetic functions available with the package include a retrieve function, a deposit function and a normalize function. All of these operations work on a consecutive three-word piece of data. This package is used to determine the heights of both triangles using the algorithm tested in equations 1, 2, and 3, and thus the coordinates (in feet), or the position of the vehicle. These coordinates are then scaled for representation on the video display. Also since the origin of the axes required for plotting is in the upper left hand corner (y increases downward and x increases to the right), the coordinates must be reversed with respect to the axes formed by the placement of the receivers. The coordinates are then in the required form for addressing the cursor of the video terminal.

These coordinates are then subjected to the expansion routine which either multiplies them by the appropriate scale factors and adds the boundary constants or leaves the data unchanged, depending on the present mode of operation (expanded or normal). If the resultant coordinates are within the bounds of the screen, they are plotted. If not, the coordinates are rejected and the bell is rung three times to signify the error. This process is then repeated.

4.J Use of the Software

4.J.1 Initialization Routine

The initialization software is quite easy to use. Once the vehicle clock has been wired to the computer interface boards, one types (while in monitor mode) 3400G. The routine will then synchronize the two clocks, set up the counting frequency to 4.8 KHz, and ring the bell every time a start pulse is received. A lower frequency counter is also initialized at 1 KHz in this routine.

To stop the routine once synchronization is finished, a manual reset is required to return the system to monitor mode.

The initialization routine is also responsible for setting the page-zero variables to their desired values for the given test situation. The locations set by the initialization routine are given in Table 2, along with their explanations.

Even though each of these variables is pre-set by the initialization routine, none of them are required for this routine itself. All are operating variables for the tracking routine. Thus, the variables can be changed in RAM before running the tracking program or, they can be changed in the code itself for a more permanent testing situation.

4.J.2 Tracking Routine

Once the vehicle is within the operating area for the system and the proper values are stored in the appropriate locations for the test site, tracking can be started. By typing $44\emptyset\emptyset_G$ while in monitor mode, the tracking sequence is begun. This initializes L1, L2, L3 to establish a working reference position for monitoring the change in L1, L2, and L3 by taking the first four good data points (all stop pulses received) and storing the most common value within a tolerance of 8 feet (plus or minus 4 feet). If no data is received, the system will get hung up here looking for data and the tracking sequence will not begin. In this event, it is necessary to reset the computer manually and start the initialization process over again.

Once data is established, the routine will check to see if an expansion character has been typed. By typing "1", the first quadrant is expanded. The screen is erased and tracking is continued on a larger scale (Figure 13). Similarly, typing "2" will expand the second quadrant, typing "3" expands the third quadrant and typing "4" expands the fourth quadrant. To return to the normal full field for the screen, a "Ø" (zero) must be typed. If the routine finds that a character has been sent, it will look for one of these characters and take the appropriate character. Any other character will be ignored and execution will continue.

From here, the routine proceeds to the averaging subroutine which in turn calls the data subroutine 4 times. Each time a receiver does not receive a transmitted pulse of sufficient magnitude to cause the counters to be stopped, the link is set, the error counter (which is cleared out at the beginning of the tracking procedure) is incremented and no data results. If this occured four times consecutively for four consecutive calls from the averaging routine, there will be no data to average and an error will occur, setting the link. The main program will then, upon finding the link set, repeat this process.

The coordinate determination routine also performs error checks on its calculations. If an error is detected in this routine, the bell is rung once and control is returned to the main program which repeats the cycle. Errors can be generated by either the math package or the various routines used by the coordinate determination routine. The only errors possible in the math package are division by zero and the attempted square root of a negative number. Division by zero can only occur if a data point is zero. This is highly unlikely since a zero point would not make it to this stage of execution. An attempted square root of a negative number, however, is more likely. If one of the received lengths is in error such that the resultant triangle does not make physical sense (e.g. the sum of two of the lengths is less than the corresponding side), then the resultant computation may lead to a negative number from which the square root must be derived. However, the data would have to be in gross error for this to occur and would be checked before this point. Thus, both of these errors are highly unlikely.

The only other computational error that might occur is in the case that the final result of the computation from the math package is too large to be represented in one word of data. This is an absurd situation since the operating field dimensions of the system are far less than the largest possible positive number representable by the twelve-bit machine. An error of this type would indicate a data error of the type that would have been detected previously. Thus, although these error checks exist for safety, they are hardly ever put to use in the computational routines.

The last error check is in the expansion routine. This checks the coordinates to be plotted to see if they are within the bounds of the screen. If not, the link is set, the bell rings three times and the main program repeats the cycle.

Thus, the most likely source of error for operation exists in the data retrieval subroutine and the interface software.

4.K Field Testing

The Passive Navigation system performed quite effectively. The position of the vehicle, determined by the time of transit of the one way pulse, employing synchronized clocks, proved satisfactory, with stabilities as predicted.

The on-shore display of vehicle position, overlayed on a memory mapped display of the pipeline position, proved to be quite effective.

The following conclusions were reached:

- For ranges in the 300 meter vacinity, under difficult multipath conditions, excellent reliability was achieved.

- Accuracy achieved was a function of hydrophone position and oscillator drift lying approximately in the one-half meter.

- Precision achieved, i.e., the repeatability of the measurement, was in the 1 meter range. In view of the multipath, this was considered excellent.

- The system permitted measurement of vehicle performance.

- The system served as a needed first step in the ultimate 3-dimensional system requested for structural navigation.

4.L Status

The homing system design philosophy with its circuit configuration and the physical construction is described in the report, "A Computer Interfaced Navigation System To allow Pre-Programmed Track Following For An Autonomous Submersible", by Franz Edson, R. Benett Bicknell, Aaron Sakovich, Michael Rubin, and William Marchant.

The system, as presently configured, is not suitable for test on the vehicle, and as a consequence primary attention was placed on the precision Passive Navigation system as discussed previously.

It is noted further that the configuration of closely spaced hydrophones provides a substantial amount of cross-coupling and shadowing. This causes errors in the assessment that may be made in the direction of arrival of the incoming signal.

In light of the inadequacies of its sensor design, as later determined, of the increasing priorities of the PNAV system in light of the structural inspection task and the tracking of its vehicle in the pipefollower task and as well, the need for redesign of the electronics, the homing system was placed on reduced priority.

Figure 8

PASSIVE NAVIGATION

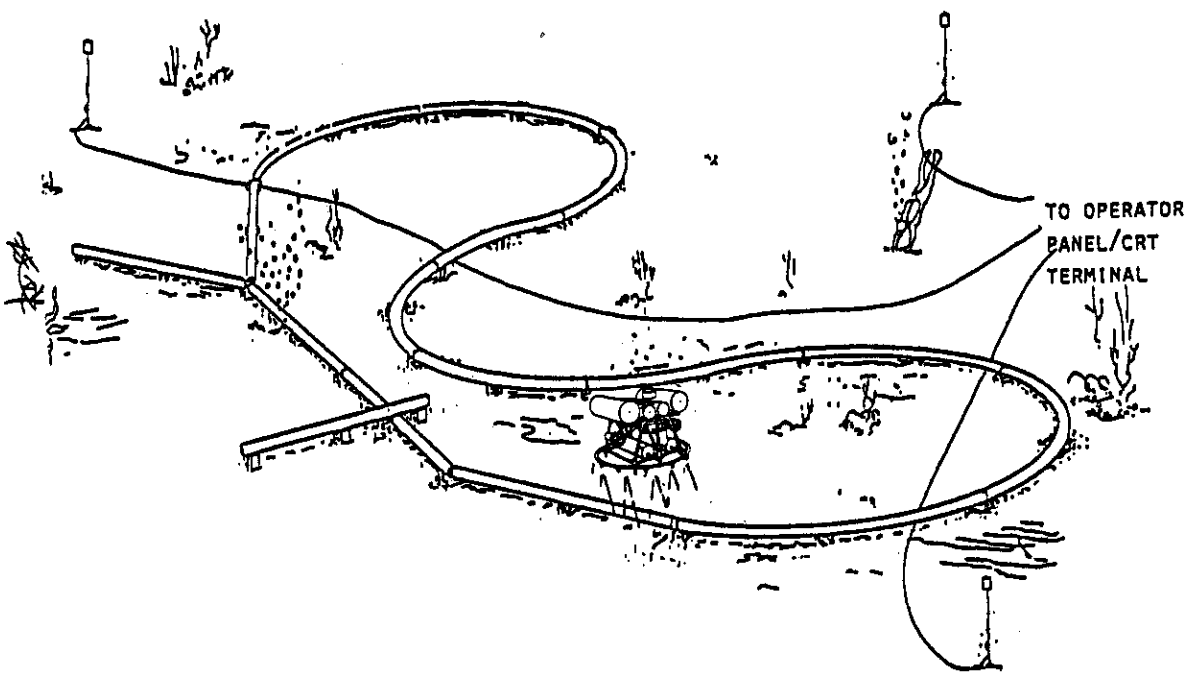


Figure 9

SYSTEM HARDWARE BLOCK DIAGRAM

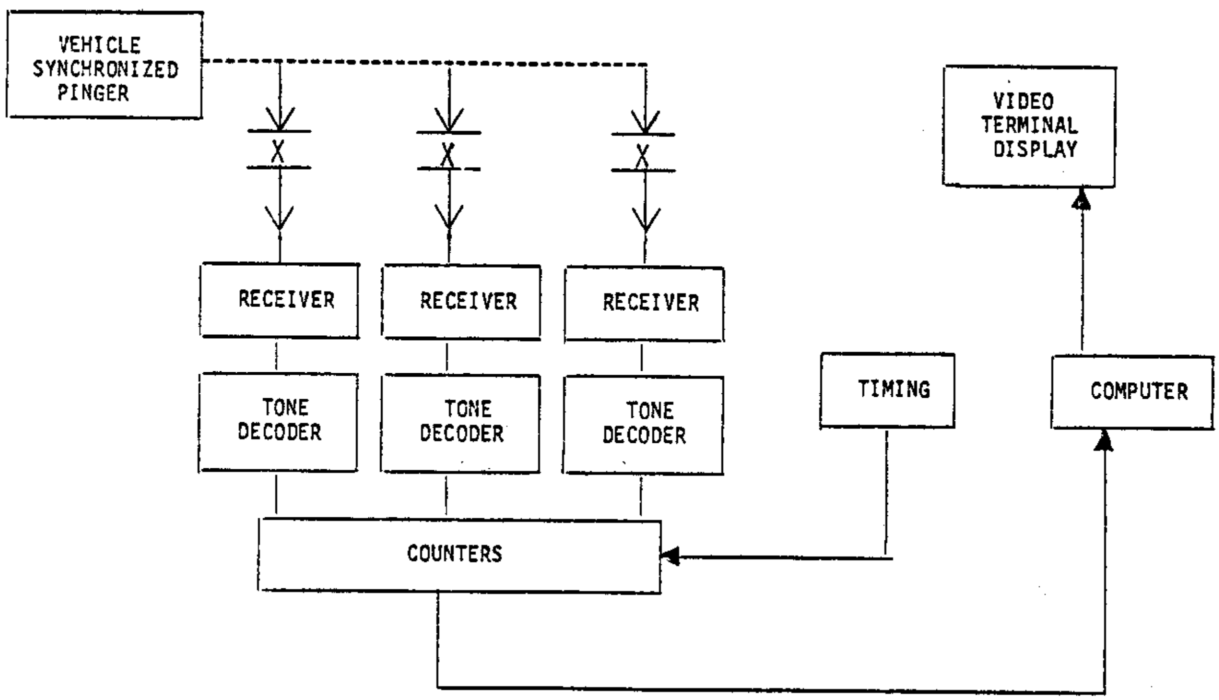


Figure 1Ø

PHYSICAL CONFIGURATION OF PASSIVE

NAVIGATION COMPONENTS

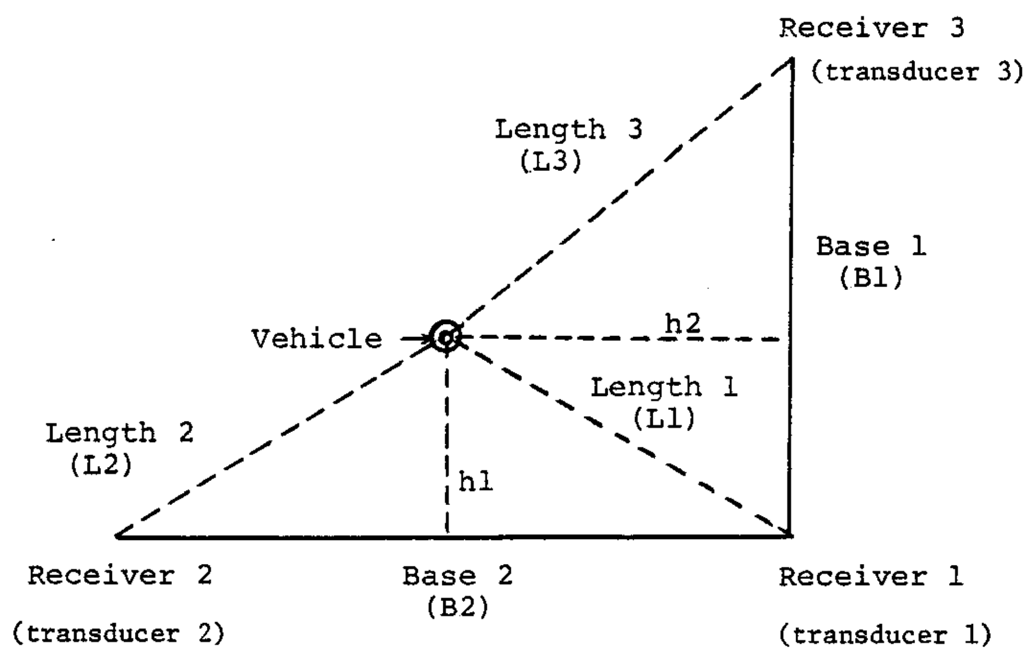


Figure 11

TRIANGLE USED IN THE APPLICATION OF THE ALGORITHM FOR THE DETERMINATION OF h

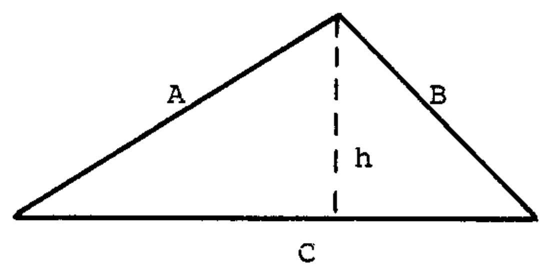


Figure 12

PASSIVE NAVIGATION OSCILLATOR

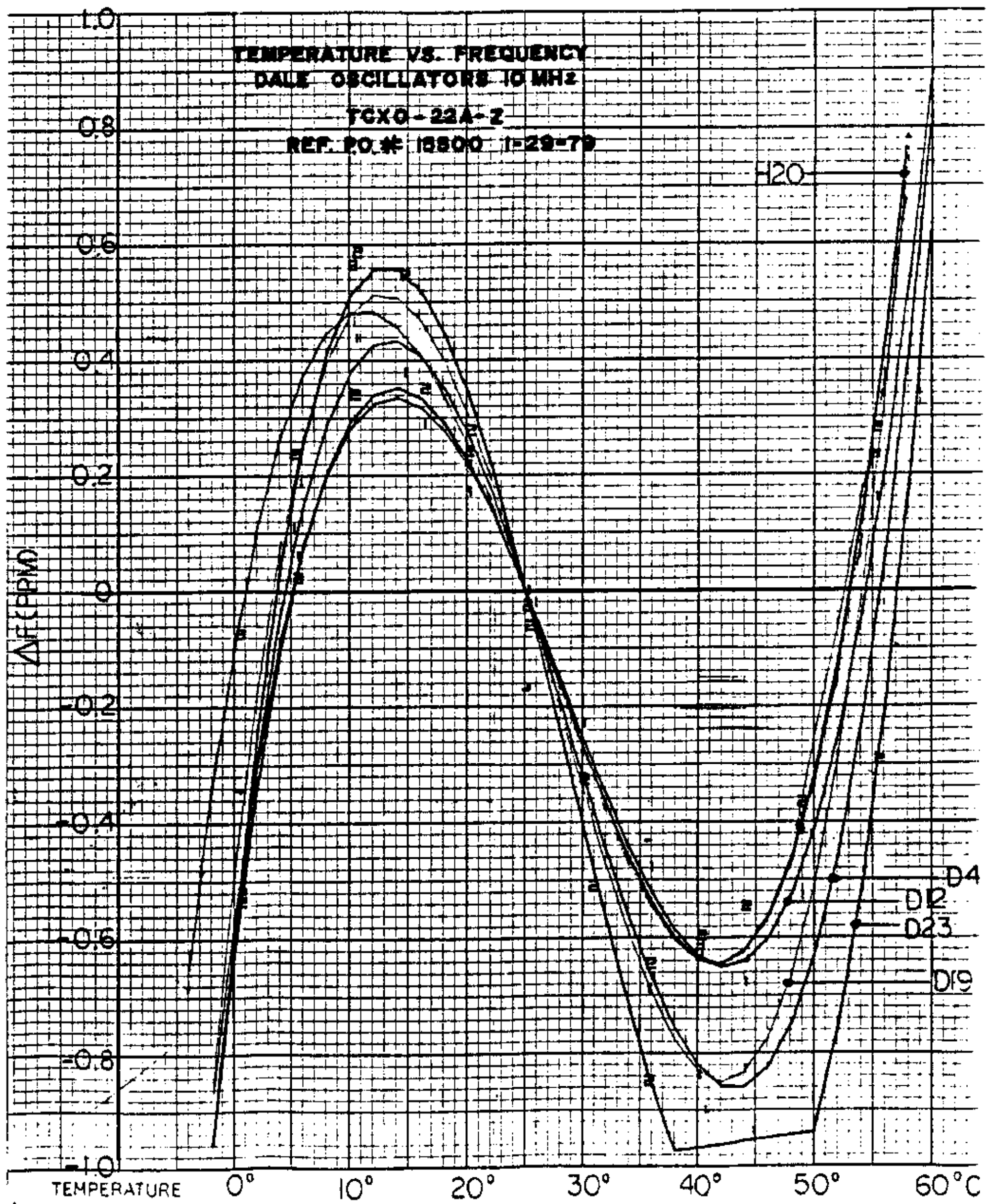


Table 2

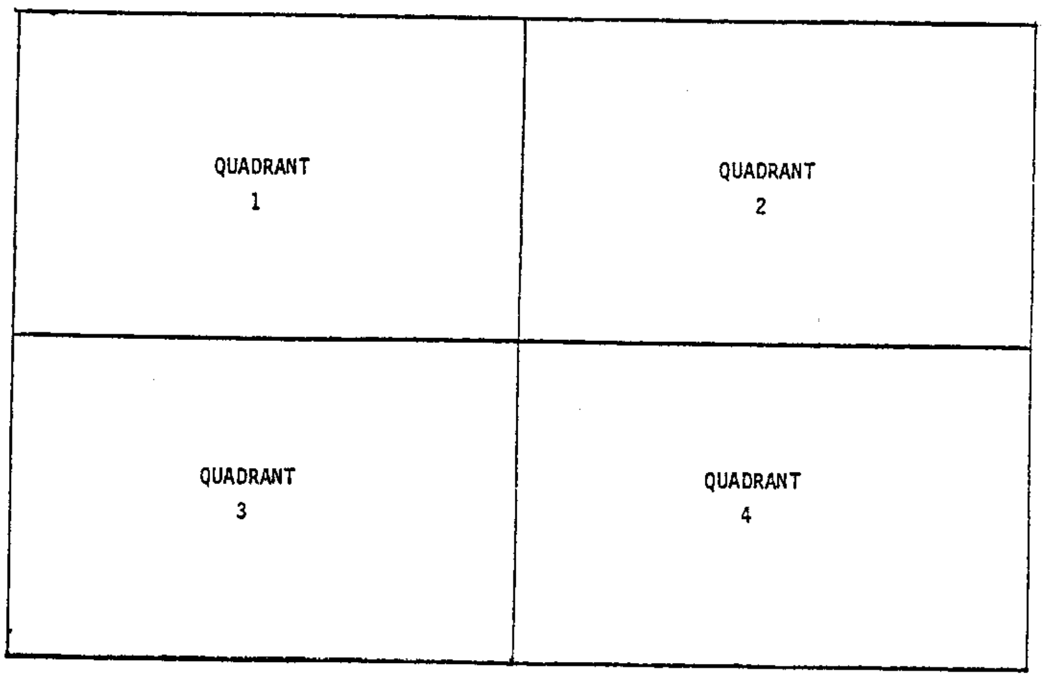
LOCATIONS INITIALIZED IN PAGE ZERO

BY THE INITIALIZATION ROUTINE

Memory Location	Variable	Explanation
0162	BASE 1	This location holds the octal representation of the length between receiver 1 and 2 in feet.
0163	BASE 2	Octal representation of the length between receivers 1 and 3 in feet.
0171 0172	COL ROW	COL is the present X coordinate of the cursor on the screen. Since a period is printed in the former location given by ROW and COL, these must be initialized. By putting 0040_8 in each of these locations, the initial period is printed in the upper most left hand position.
0151 0152 0153 0154	MCSCL MRSCL COLSCL ROWSCL	These are scale factors which must be used in scaling the coordinates down to fit in the field of the video screen. By using the values for BASE1 and BASE2 for ROWSCL and COLSCL respectively, and using the octal dimensions of the screen, 30_8 by 120_8 (24_{10} by 88_{10}), and exact scale is formed.
0177	CORRCT	This variable was used as an additive factor to compensate for phase differences between the shore clock and the vehicle clock. The problem has since been, for the most part, corrected and so the location is initially clearly out.
0176	UCHK	The PNAV system is currently set up for optional use with another terminal to be used for printing each data set and each data set average. It is used solely for testing and the option will be eliminated from the finalized system. Setting 7777_8 in this location activated the option and clearing this location disables the option.

Figure 13

SCREEN REPRESENTATION



5 THE ACOUSTIC LINK

5.A Introduction

The conventional tether supplies a very effective communications channel between submersible and the surface suport facility. This channel must be replaced, if the vehicle is to be free of the tether. The EAVE team has chosen to work with acoustic technolgy as the replacement, as opposed to electromagnetic communication, which was rejected for substantial reasons.

For reasons similar to those noted in the previous discussion on navigation, the demands placed on the communication link are directly related to the assigned mission, which in turn establishes boundaries on acceptable bandwidth, error rate and range. There is no universal communication system, and the initial system design most properly should be pointed toward vehicle support.

It is vital to note that a major consequence of a computer system onboard the vehicle is a reduction in the demand for command and control bandwidth, and potentially in the acceptable error rate. No longer need the operator fly the vehicle in the sense that he need control all functions. The sensor systems - navigation, depth and velocity inputs, combined with onboard mission programming join to make the vehicle self-sufficient.

The operator provides overriding judgement to assure that the program is meeting his objectives. He applies his commands in terms of objectives, with the consequence that the demands on the command and control channel are reduced. The specifications on the required acoustic channel, given full microprocessor control appears to be substantially relaxed. It is apparent, of course, that in many missions wide band data links may still be required, for example to transmit television pictures. Important work is being done to transmit visual images over acoustic links, with promising areas of technology still to be applied. The implications are yet to be examined in this program.

The EAVE Project Team has established objectives for the initial communication system development. We wish to remove the tether now used to load mission instructions into EAVE, and thus to further the development of an autonomous vehicle. We wish further to employ the communications system as a means of evaluating the performance of the vehicle at the Lake Winnipesaukee Test Facility. Due to financial constraints, the initial hardware has the purpose of extending the vehicle, and not to demonstrate the ultimate in communications.

The technical objectives of the system that replaces the hardwire Teletype link, now used to load EAVE programs prior to launch, and to permit communications with the vehicle while under autonomous testing, include the following:

Data Rate:	110 Baud
Range:	200 meters
Error Rate:	1 in 10^4
Water Depth:	10 meters

The initial step is to transfer command and control data to the vehicle from the operators station. Subsequent development will include two-way communication.

A communication link has been fabricated which will allow communication between a remote operator and the vehicle computer system. The link will permit the vehicle to report status to the operator while allowing the operator to intervene, or to change the vehicle's instructions. The system employs a transmission scheme which uses nine frequencies centered around 28 kHz. These nine frequencies are transmitted (a logical 1) or not transmitted (a logical 0) in sequence. The receiver then receives and decodes this nine bit data word. The data, which is to be received or transmitted, is defined as one of three types and is encoded accordingly by one of three methods. The first type of data is considered to be of the highest priority (control instructions). It is transmitted by the operator and received by the vehicle. The received data is then re-transmitted by the vehicle to the operator so that the operator's computer can verify that the data word is the same as it sent. This method is a type of "hand-shaking". The second type of data is a middle priority type (status information). This data is transmitted after being encoded according to an error correction and detection algorithm (hamming code). The received data is decoded by the receiver using the same algorithm. The third type of data is of lowest priority and is transmitted and received directly without any coding or decoding.

5.B Acoustic Transmission In The Water

Water is, in principle, a very good medium in which to transmit acoustic energy. At 28 KHz and over short ranges, attenuation is modest and is largely a function of the spreading loss. In shallow water the medium is bounded by reflecting surfaces and is often filled with reverberators.

The received signal, therefore, is typically a combination of several signals, the main signal and the unwanted reflections. Because the path length of the reflections are longer than the main path, the reflections arrive delayed in time and somewhat attenuated from the main signal. Discrimination of the direct path signal, which differs from most echoes in amplitude and arrival, is often difficult.

The multipath sometimes lasts long enough to cause unwelcome delays in the transmission of subsequent pulses, thus limiting the data rate of the channel.

5.C Addressing the Multipath Problem

The system addresses the multipath problem by transmitting tones that are discrete in time and frequency. The transmitter sends out 9 serial tones, each tone corresponds to a data bit. The lowest tone is 23.44 KHz and the highest is 31.25 KHz, with 7 tones evenly spaced in between.

The first tone sets up a timing sequence in the receiver, once this is done, the receiver looks for particular frequencies at a particular time after the first pulse. The presence or absence of a tone is determined by tone decoders.

Each tone pulse has its frequency spectrum spread or smoothed by the sinc* function because each tone has a rectangular pulse shape. The pulse width T is 1/110 sec. The corresponding frequency spread between the first 0 crossings of the sinc function is 2/T = 220 Hz. See Figure 14.

---

*A sinc function has a Sin X/X distribution

The total signal of nine tones which consist of 9 of these spectra whose center frequency varies with each tone.

From a frequency spectrum point of view, there is little overlapping of pulse energy.

When little or no multipath exists, it is fairly easy to detect this spectrum. Multipath tends to distort the amplitude and phase of the spectrum by constructive and destructive interference. There is also some frequency shift due to the doppler affect caused by moving surface waves, but this is small and can be ignored with the system bandwidths employed.

5.D Acoustic Link Hardware

The acoustic link hardware was designed to be used as an add-on module to a standard modular 6100 microprocessor board. The signals required to drive the hardware are all obtained from a 6402-based UART. The microprocessor system consists of 2K of PROM, 1K of RAM, a CPU, and 2 UARTS. One UAR]OT is used to communicate to a terminal via a standard RS-232C link. The other UART is used to drive the acoustic link hardware. The acoustic link hardware has two separate sections, one for generating and transmitting acoustic signals, the other for receiving and decoding them. The hardware requires no special consideration from the microprocessor, for it is not oncerned whether its imputs come from a RS-232 link or an acoustic channel.

5.D.1 Acoustic Transmitter

The acoustic transmitter accepts data in serial form from a UART, along with timing signals. The timing signals are counted in a manner which is suitable to control a digital phase-lock loop. The frequency of the phase-locked loop is increased synchronously with each serial bit from the UART. That is, the digital phase-locked loop generates a separate tone for each data bit.

The tone is then keyed (gated on or off) by the data that is present. The acoustic signal is now an amplitude modulated signal with a stepped carrier frequency. This acoustic signal drives a power transmitter which puts out 20W of continuous power. This transmitter is flat out to 50 KHZ. The acoustic power drives a 27 KHZ transducer that has a Q of 3.

5.D.2 Acoustic Receiver

The acoustic receiver hardware consists of a receiver amp, a tone detection stage, and timing reconstruction hardware. This hardware also requires timing signals provided by a UART, and generates serial output data suitable for reading by a standard UART.

The receiver has 60 DB of gain, is AGC controlled and has 40 DB of dynamic range. The receiver is bandpass filtered around the center frequency, with a bandwidth of 10 KHZ. The output of the receiver is a 1 volt acoustic signal which is fed into a series of tone decoders.

All nine tone decoders continuously receive the acoustic signal, that is, the output of the receiver amp is not gated on or off in any way.

The tone decoders feed a demultiplexer. The clock signals for the demultiplexer are generated from a counter which is keyed by the leading edge of the first tone. Once the counting sequence has started, the counter is held off from being reset until the full word length is counted out. This prevents a multipath signal of the start tone from restarting the counter in the middle of receiving a word.

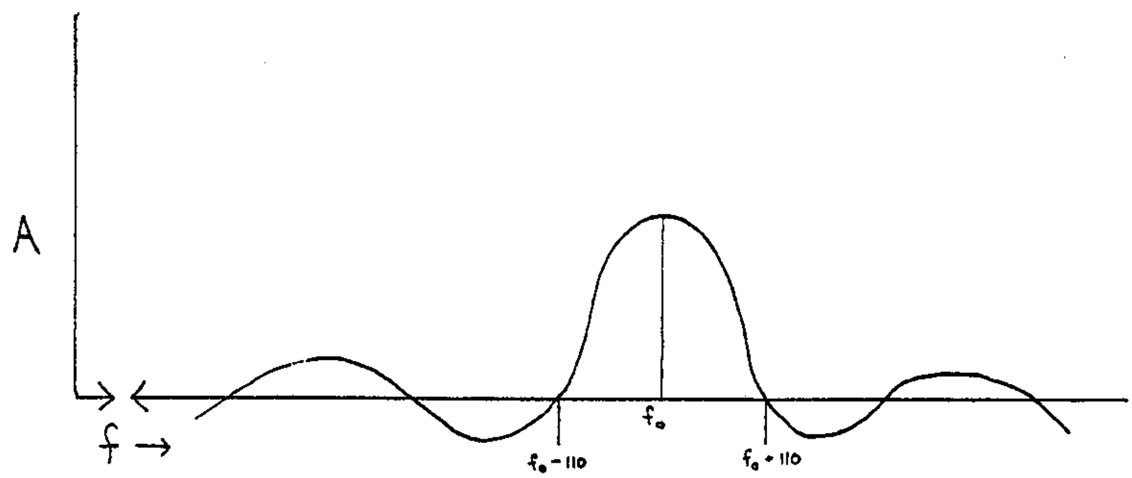
The tone decoders used are the EXAR 2212 components. They have approximately 1 KHZ bandwidth each, and there is one tone decoder for each of the 9 tones transmitted. The demultiplexed output of the tone decoders is reconstructed serial data which can be read by any standard UART.

5.D.3 Acoustic Link Software

A special application package of software must be included for use with the 6100 system. It uses the intelligence offered by the microprocessor to perform any coding and decoding of data passed over the link, as well as error detection and correction. The software also contains the protocal required to manipulate the transmitter when more than one transmitter uses the same channel. All of the code required for the link takes up less than 500 words of memory.

Figure 14

AMPLITUDE vs. FREQUENCY FOR 9 BIT WORD



Frequency Spectrum of a Single Tone

6. Field Testing

In the summer of 1979, the EAVE vehicle was taken to the UNH test facility in Lake Winnipesaukee, NH for evaluation after a winters redesign. This facility described in earlier reports, is 10 to 20 meters deep. A simulated pipeline was laid on the lake bed, and a barge, containing electronic and service facilities, was stationed near-by.

The objective of the test was to demonstrate fully autonomous vehicle operation under internal software control, using the data generated by on-board sensors to acquire and to follow the pipeline. Elements of the Navigation and Communications subsystems were also tested.

The objectives of the test were fully met. The vehicle performed numerous maneuvers under software control, changing elevation, circling buoys and traversing lines. To be fully autonomous, however, the vehicle must also sense its environment, and must respond properly to it. In these tests the only sensors carried on-board were the acoustic transponders designed to perform the elementary mission of pipeline acquisition and following. In the tests the vehicle did indeed acquire and follow the pipeline successfully. It satisfactorily performed to the full capability of the hardware and software that it carried. Underwater motion pictures taken at the tests, demonstrated the success of the development, and provide a record that will permit further design improvements.

Although success was encountered, and the vehicle operated in a totally autonomous mode that justified the intensive effort that went into it, attention must be given to the learning that resulted and to the deficiencies that exist in the system. The fundamental objective of the EAVE program is to develop technology, and not to build a vehicle for a specific mission. With this in mind it is necessary to review the lessons of the past summer's work. These points include:

a) Once again we learned that nothing occurs automatically in the development process. The careful work of the winter months was often found to be deficient in the field tests in the summer. In particular, the integration of software and hardware is difficult to achieve, for engineers and programmers operate on different conceptual levels. This learning is not new but once again it demonstrates the increasing complexity of ocean systems engineering.

b) The acoustic system of 12 sensors, spaced 30° around a circle is divided, by the software, into four quadrants for vehicle control and maneuvering. This adequate control to achieve the results of this past summer. It was found to be too coarse to be used in future tests. At the time of its design, the memory available in the vehicle's computer was too limited for a more sophisticated system. This constraint is now removed, and the software to achieve more subtle control with 12 quadrants, as well as the more sophisticated maneuvering routines that are obviously needed, may now be installed. The fundamental software concepts that evolved for the EAVE pipeline following mission were proven to be reliable and effective. Their extension to make a more competent pipeline follower is quite reasonable to implement if it meets the needs of the program.

c) As an intermediate step to achieve improved control at a lesser cost in software, several hardware changes have been suggested, including a reorientation of the sensing transducer ring.

d) Substantial improvement was observed in the performance of the thrusters, in the reliability of the sonar transducers, in the reliability and effectiveness of the electronic circuitry and in the battery charging system. These were all design goals of the past season.

- EAVE now maneuvers with five degrees of mechanical freedom, permitting it to be considered for a wide range of potential inspection mission. With this excellent maneuverability comes the potential for extensive cross-coupling of response and the possibile need for a sophisticated control system involving far more than the rate damping employed in EAVE. Examination of the films of the vehicle tests reveals that it was heavily damped, and that the speed of response, called for by the sensing system was slow. Suspicion of some cross-coupling of responses remains. This matter, however, may be examined analytically, and modified control equations may be developed. In future tests, the response of the vehicle, and the design of its control system clearly must be studied.

- The navigation system was set in place at Lake Winnipesaukee and operated as a shore-base vehicle location system, as described in Section 4. The hardware, including displays, worked effectively at the end of the season, although the navigation system was not integrated with the vehicle system in 1979, and the potential for autonomous navigation was not tested.

- The communications system underwent component testing as described in Section 4. No systems integration was attempted.

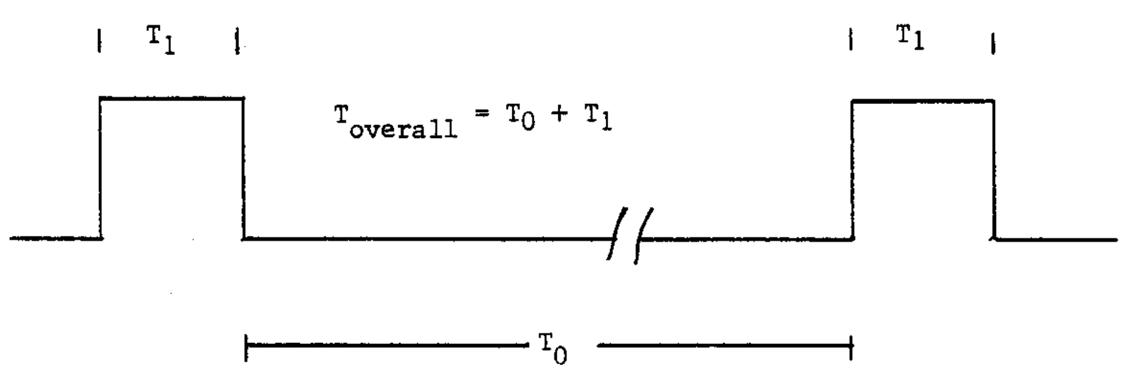
APPENDIX A

TIMER BOARD

The timer board consists of a central 10 MHZ oscillator, two programmable divide by n counters, several fixed counters, latches, and fa P.I.E. The timer board generates 3 output signals. One is a symetrical square wave at 10 HZ. The other two are programmable via the computer.

The two programmable timers have non symetrical outputs. The waveform is a pulse with fixed logical one duration (T_1) and variable logical 0 duration (T_0).

PULSE DIAGRAM



Each programmable timer has different operating limits. There is a high and low frequency timer whose range of possible frequencies overlap. The high frequency timer has a frequency range from 100 KHZ to 60 HZ or $T_{overall}$ = 10 μsec. to 16.66 msec. T_1 for the high frequency timer is 1 μsec.

The low frequency timer has a frequency range from 100 HZ to sec.

The frequencies obtained from the timer are integer divisions of the fundamental.

i.e. $$f_{out} = \frac{f_0}{n} \text{ where } n = 1,2,3...1667 \qquad (1)$$

or

$$T_{overall} = N\ T_0 = \text{ where } n = 1,2,3...1667 \qquad (2)$$

f_0 is 100 KHZ for the high frequency timer (T_0 = 10 μsec) and f_0 = 100 HZ for the low frequency timer (T_0 = 10 msec.).

The integer n is a binary integer stored in location 101 for the low frequency timer and in 102 for the high frequency timer. The integer n is a twelve bit binary number broken down into three 4 bit fields. One field corresponds to the count length of one decade. The lowest four bits (DX11 - DX8) govern the 10^0 count. The middle bits (DX7 - DX4) govern the 10^1 count, and the highest bits (DX3 - DX0) govern the 10^2 count.

EXAMPLE 1: Produce pulses at a frequency of 813 HZ.

Using equation 1,

$$813 \text{ HZ} = \frac{f_0}{n} = \frac{100 \text{ KHZ}}{n},$$

$$n = \frac{100 \text{ KHZ}}{813 \text{ HZ}} \qquad 123_{10}$$

or for the programmer

	$\underline{10^2}$				$\underline{10^1}$				$\underline{10^0}$				
DX	0	1	2	3	4	5	6	7	8	9	10	11	
DATA	0	0	0	1	0	0	1	0	0	0	1	1	$= 0443_8$

It is important to note that the field counters may be loaded with a number greater than 9μ greatly extending the range of the counter.

EXAMPLE 2: Produce a pulse with a period of 12_0. This can be accomplished two ways.

a)

	$\underline{10^2}$				$\underline{10^1}$				$\underline{10^0}$				
DX	0	1	2	3	4	5	6	7	8	9	10	11	
DATA	0	0	0	0	0	0	0	1	0	0	1	0	$= 0022_8$

or by loading binary equivalent of 12 into 10^0 register.

b)

	$\underline{10^2}$	$\underline{10^1}$	$\underline{10^0}$	
DX	0 1 2 3	4 5 6 7	8 9 10 11	
DATA	0 0 0 0	0 0 0 0	1 1 0 0	$= 0014_8$

The maximum count for each field is 15, there the maximum count is: $15 \times 10^2 + 15 \times 10^1 + 15 \times 10^0 = 1500 + 150 + 15 = 1665$

Operating the timer board is simple. The number n is loaded into the low and high frequency timer latches via the write 1 and write 2 commands.

After the number is in the latch, the number is strobed into the timers and the timing begins by setting Flag 1 and Flag 3 for the low and high frequency timer respectively.

POSSIBLE SOURCES OF TIMING ERROR

One source of timing error is inherent to the 4059 counter chip. Whenever a new number is loaded into the counter, or whenever the counters are initialized, the period of the first count cycle is increased by an amount of time equal to its T_0. Thus:

$$t_{overall} = (n + 1)\, T_0 \qquad \text{Eq. 3}$$

This is only for the first count cycle, after the first pulse is generated, the period of the pulses is just n times the T_0 of the counter.

This does not introduce a synchronization problem between two counters if both counters are started simultaneously. This is because both counters will have their count length increased by one time T_0, and then they will both go to their normal count cycle.

When a new number (n) is entered to change the count length, all of the counters for that timer (ie. low or high frequency) are reset. Thus, the only error introduced from the initialization process comes from the uncertainty of the location of the highest frequency. In the worst case, this would be one period of the highest fixed frequency into the individual timers. In both cases this is 10 MHZ. This gives a maximum initialization error of .1 μsec, between two counters started with synchronous signals but driven by asynchronous clocks.

The code that operates this clock follows in the program listing. After this code is entered, the numbers n_1 for the low frequency clock and n_2 for the high frequency clock are stored in location 101 and 102 respectively. Typing 600G will initialize and start the timers, counting by the number n. This is just representation coding and by no means is absolute. Programming code for this clock is simple enough so that it can easily be written for specific applications and locations in memory.

APPENDIX B

UART DESCRIPTION

The 6100 communicates with the operator via a universal asychronous receiver/transmitter (UART), see Figure 15. The principle element on the UART is the Intersil 6402 UART chip. This chip can accept and transmit serial and parallel data bit stream. The Intersil 6101 parallel interface element (PIE) is used for some of the timing and control but basically for communication to and from the CPU. This board accepts a parallel 8 bit word, formats it (1 start bit, 8 data bits, 1 stop bit) and transmits the resulting 10 bit word serially through a 20ma current loop. Operating at the same time, a serial to parallel receiver is monitoring a 20ma current loop till it senses a word being transmitted to it. It strips off the start and stop bits and signals that it has completed the task. These two modes of operation (receiving and transmitting at the same time) form a full duplex link to the real world.

There are other features to the UART that make it a very flexable tool for transmitting and receiving. One such feature is a variable speed selector. This allows data to be transmitted/received at different rates depending on the speed of the terminal used. The operator can select (via switches on the UART, see Table 3) a speed at which to run the terminal. Later on, the speed may be altered by software if a different terminal is used. This type of adaptability is necessary because terminals (and peripherals) can accept data at different speeds. For example, if communication between two computer systems is desired, a very high data rate can be used. However, if an ASR33 Teletype were used, it can only receive and transmit at 110 baud. The Harris 6440 baud rate generator element is used here. This chip allows for the selection of 1 of 16 possible clock rates including a user defined one.

Also available to the systems designer are all the timing and control lines that the UART uses. (ie. the baud rate clock, transmissions completed status, busy status, start transmission, word being received status, to name a few). Therefore, if a design called for the transmission of data by means other than a current loop or an RS-232 standard (which will be discussed in the next paragraph) the necessary timing and control will all be available.

The 20ma current loop transmission of data has some significant advantages. Most ideally for our purposes, it provides a highly reliable link over long distances. In other words, noise affects a 20ma current loop far less than it might a RS-232 link. (Noise might have infinite power at one instant in time but it has very little average power over an interval. So, the noise cannot drive the current loop but it might give spurious voltage levels to the RS-232). With this 20ma current loop, a translator circuit has to be used to switch a 20ma loop to an RS-232 standard and vice versa. Our design of the translator also incorporates opto coupling, thereby eliminating ground noise from the terminal and its power supply.

In all, the UART is a highly reliable and flexible tool. It can supply the user with many operating speeds, transmission modes (20ma or RS-232) and control and timing for many communication applications.

IMPLEMENTING INTERRUPTS ON THE UART

I. References:

a. IM6100 Microprocessor Handbook

1. Priority for Vectored Interrupts (p.47-48)
2. Interrupt/Skip (p.48)
3. Control register A formatting (p.48)
4. Interrupt Handling (p.51)
5. Device Interrupt Grant Timing (p.19)
6. CP Interrupt Transfer (p.20-21)

b. Intersil Change Notice of April 11, 1977 (p.13-20, P.11-12)

c. Intercept Jr. tutorial - Owner's Handbook (p.2-14)

II. Interrupt Program Set Up

Location	Contents	
0000	0000	/location of where in program interrupt occurs
1	6002	/turn off interrupt system
2	RETURN	/return to monitor (should never reach here)

III. This next routine should be located in the monitor to initialize the UART. It loads the vector register as well as Control registers A and B. The baud rate is initially set up to be 600 baud but can be changed via the program in the Baud Rate Selection routine listing. (Located later in this report).

UART INITIALIZATION ROUTINE

```
XXXX        Sense 1 handler     (should be Jmp instructions)
XXXX+1      Sense 2 handler     NOTE: Jump instruction table to
XXXX+2      Sense 3 handler     handle the vector register
XXXX+3      Sense 4 handler     return (ie. if the vector reg-
                                ister contained 0200: when an
                                interrupt occurred, on say sense
                                3, the CPU will execute the in-
                                struction located at 0202, with
                                sense 1 the instruction at 0200
                                will be executed and so on.


UART Initialize, IOT RUN        /IOT RUN = 6407
                 ION            /Reset CP FF after (6001)
                                /Instruction
                 JMPI .+1       /Go to start of initialization
                 STRT           /Sequence

STRT,            IOF            /Initialization Routine (6002)
                 CLA
                 LDA CRA        /Load Control Reg. A word
                 WCRA           /Write to UART (WCRA = 6165)
                 CLA
                 LDA CRB        /Load Control Reg. B word
```

```
WCRB            /Write to UART (WCRB = 6175)
CLA
LDA VR          /Load Vector Reg. word
WVR             /Write to UART (WVR = 6174)
CLA             /
CAF             /Clear all flages (6007)
JMP Program     /Jump to other programs
```

CRA = 7201 (Number depends on specific interrupts used)

CRB = 0067

VR = XXXX (See note above)

When the interrupts are required, insert a ION instruction. Don't forget to reactivate the interrupts in your service routines.

BAUD RATE SELECTOR PROGRAMMING (Via Software)

1. Select the speed desired from Table 3
2. Load the speed into the accumulator
3. Strobe the UART Write 2 pulse (6171)
4. Change speed on the terminal

Recommend that this be implemented into a monitor command (ie. 600S (change speed to 600), 2400S (change speed to 2400))

BAUD RATE SELECTOR VIA HARDWArE SWITCHES

1. Select switch combinations from Table 3
2. Press reset button (half way running program)
3. Set the switches on the UART (UP or DN)
4. Change speed on terminal
5. Press reset button

Baud switches are set to run UART at 600 Baud and the software select Table 3 is adjusted for this. If the switches are changed, this table is not valid.

Figure 15

UART

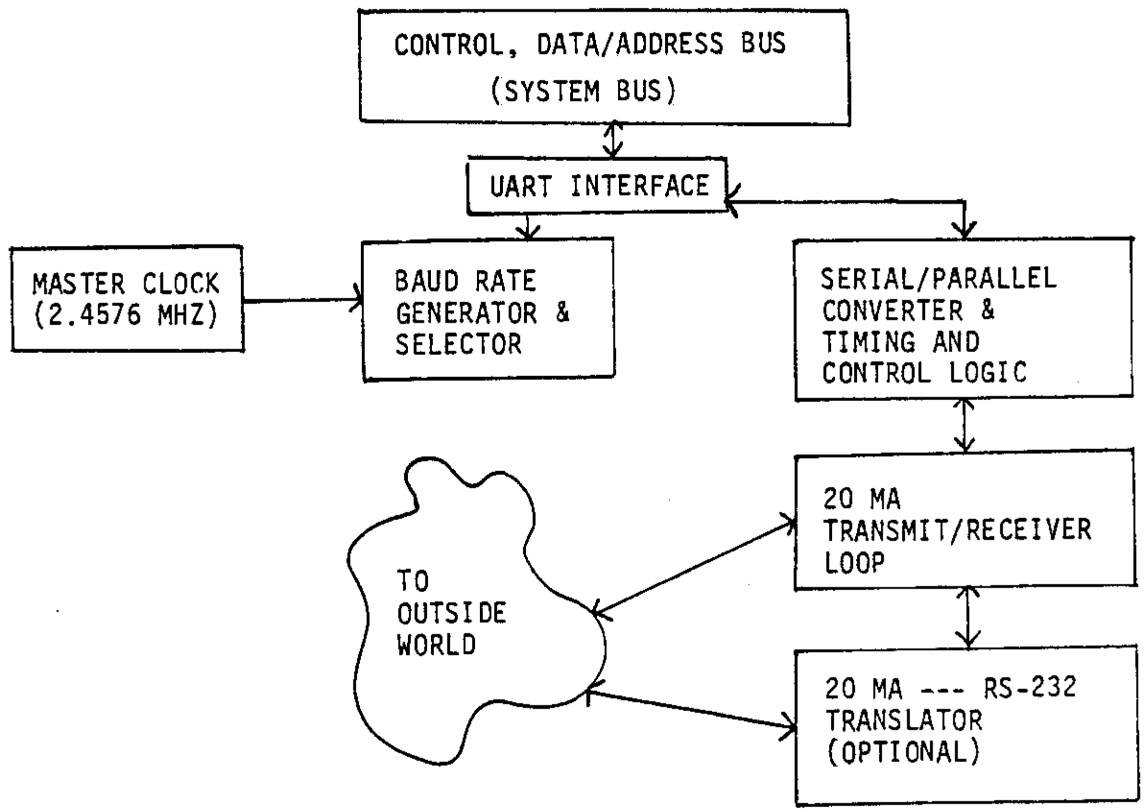


Table 3

BAUD RATE SELECT CODES

SPEED DESIRED	(IN OCTAL)	S1	S2	S3	S4
MIXED INPUT (I_M)	0006	DN	DN	DN	DN
2000 BAUD	0007	DN	DN	DN	UP
50 BAUD	0004	DN	DN	UP	DN
75 BAUD	0005	DN	DN	UP	UP
134.5 BAUD	0002	DN	UP	DN	DN
200 BAUD	0003	DN	UP	DN	UP
600 BAUD	0000	DN	UP	UP	DN
3600 BAUD	0001	DN	UP	UP	UP
9600 BAUD	0016	UP	DN	DN	DN
4800 BAUD	0017	UP	DN	DN	UP
1800 BAUD	0014	UP	DN	UP	DN
1200 BAUD	0015	UP	DN	UP	UP
2400 BAUD	0012	UP	UP	DN	DN
300 BAUD	0013	UP	UP	DN	UP
150 BAUD	0010	UP	UP	UP	DN
110 BAUD	0011	UP	UP	UP	UP

APPENDIX C

PROM BOARD

(PROM Programmer)

Although flexibility of the vehicle system can be lost when permanent memory is used in the system, there is much to be gained by writing some of the software components into semi-permanent memory. To accomplish this, a 'Prom' printed circuit board was developed. This computer component consists of 2K by 12 bit UV erasable prom with the decoding circuitry able to decode 1 of 3 1Kx12 blocks directly. The board uses IM6653 1024x4 UV eproms developed by Intersil.

In order to program the proms, a programming board was developed which would allow programming of 1Kx12 blocks of memory with some error detecting capability. The following pages describe the programmer operation.

PODT

Prom Burning Monitor

Introduction:

The five original files of ODT:

PRODT.PAL

CALLI.PAL

PRIPRM.PAL

PINTJR.PAL

CHARDR.PAL

These have been combined into one file: PODT.PAL.

PODT includes the following features, which are documented in detail herein:

- Breakpoints (B)
- Selectable UART baud rate (S)
- Lengthened Command Scanner that recognizes:
  - I,P,R,L,B, And S.
- Linkage to PROM burning software. (I,P,R)
- Command for loading RAM from TI (L)

The prom burning software is in file: PROM.PAL, and has been located to fit along with PODT in the monitor EPROM in 6-8K. It is simply accessed using the command scanner of PODT. Additionally, a program for initializing RAM to all 7777 is included.

These two files were burnt into prom and labeled PODT on 7/29/79. Three sets were burned.

The files PODT.PAL and PROM.PAL are available on dectape VID:900505 until 9/1/79.

Breakpoint Feature:

The breakpoint operates only during program execution, ie. by typing XXXXG. The G handler places a trap instruction in the location where a breakpoint is specified. When the trap instruction is encountered, the original contents of the breakpoint are restored, a message of the form:

AAAA B - CCCC DDDD

is sent to the terminal, and control returns to ODT, at the point of waiting for a new command.

AAAA - Breakpoint Location

B - Link

CCCC - Accumulator

DDDD - MQ Register

<u>Breakpoint Specification:</u>

The breakpoint address is stored in TRAD = 30 (Trap Address).

1. During ODT initialization, the breakpoint is set to 27 = OBPLOC (original breakpoint location)

2. By typing, XXXXB, the breakpoint may be set at XXXX.

3. Typing just B, restores the breakpoint at OBPLOC.

Two important facts:

1. If a program (ie. XXXXG) does not encounter the trap instruction, but returns to ODT by either program flow or operator reset, the operator will have to remove the trap instruction and restore the original contents, which are stored in KEEP = 31.

2. Breakpoints can only be set in RAM.

Selecting UART Baud rate From the Terminal

The UART speed is set to the speed determined by the position of the 4 switches on the UART board, during ODT initialization. To change speed of the UART from the terminal, type XXS.

XX is the octal number for the four bit binary number which is determined in the following way. Compare the switch positions for the new speed to that of the old if the switch must change positions set $S_n = 1$ if not $S_n = 0$.

$$XX_8 = S_1\ S_2\ S_3\ S_4)_2$$

Example: With the UART set at 600 baud, typing 13S will change the speed to 300 baud.

Note that Table 3 gives the values of XX for a UART set at 600 baud during ODT initialization.

WARNING: S IS RARELY ECHOED TO THE TERMINAL.

PROM Burning Using PODT

The following steps are followed to burn EPROMS. See the original PROM burner documentation for explanation of software and error messages.

1. Initialize RAM to 7777 before loading with code so that unused locations are not needlessly burnt.

    a. Set limits of initialization in page zero.

        101 - Starting address to be initialized

        102 - Last address to be initialized

    b. Type I

2. Load RAM with data to be burnt.

    If from TI terminal, type L, and then press fom: (rewind playback), (playback on).

3. Specify the following:

103 - Location in EPROM to start burning (0-1777)

104- Location in RAM to get first data to burn into EPROM.

105 - Location in EPROM at which burning stops (103, 105, 1777)

4. Type R to read EPROM from 103 to 105 to be sure they are erased.

5. Turn on -40^V supply with V=0 and slowly turning to -40^V.

6. Press F to burn (FIPE)

APPENDIX D

TEST OUTLINE RESULTS (SUMMER 1979)

The EAVE vehicle has undergone extensive modifications since its summer tests of the previous year. Modifications and changes were made to the buoyancy tanks, battery cases, and thrusters. Each of these had to be tested in the working environment before the prime mission of pipefollowing could be started.

The first several days were spent conducting pressure, buoyancy, stability, drag, and thruster tests. The pressure tests required that the vehicle be sunk to the deepest part of the test area (approx. 150'). It would spend approximately 1 hr. at this depth and then taken aboard the barge. A thorough examination would then be conducted for detection of any leaks in the pressure cases, battery housings, or buoyancy cases. A shallow water pressure test would then follow (depth of approx. 30 ft.) and the vehicle would then be re-examined.

The vehicles' weight and hydrodynamics have been altered such that new experimental data had to be taken to better define the vehicle's operating characteristics. The determination of the vehicle's buoyancy drag and stability had to be determined. The buoyancy would be determined by placing the vehicle in the water and remove all tethers. (Note: The vehicle would contain all necessary hardware that it would normally carry). Weights would be added on to the vehicle till it would start to sink. The added weight would determine the amount of positive buoyancy stability criteria would be dependant on how the vehicle 'sits' in the water with no outside interference (ie. thrusters turning on or off). Also asked was the effect on vehicle leveling when forces were applied to different areas of the vehicle.

Drag experiments were planned to demonstrate how much resistance is being generated when the vehicle tries to perform a maneuver.

The thruster tests examined the amount of thrust each motor can generate. In addition to documenting the thruster performance, the mission abort timers, that allow the vehicle to be recovered in case of hardware or software malfunctions, were tested.

The final piece of hardware to be tested was the sonar system. The tests on each of the twelve transducers included checks for line noise, stability, and reliability of the incoming data. Also, tests were conducted on all twelve transducers simultaneously to check for proper program sequencing, signal overlap and uniform readings.

This concluded the testing of the hardware of the EAVE vehicle. It was then necessary to examine the hardware under software control. The vehicle's auto altitude routines were tested for proper hovering height above the bottom. Lastly, the pipefollowing software itself was tested, checked, and executed with the tether attached and detached.

PRESSURE SEAL TEST

The vehicle had been modified over the winter to include several new pressure cases and tanks. The cases had had only minimal testing for seal leakage because of lack of sufficient indoor test facilities. Therefore, the pressure seals on all of these cases had to be tested at depth.

The vehicle was lowered to a depth of 10 feet for approximately 30 minutes. Observations showed no leakage in either battery case and no leakage in the primary electronics housing. However, slight leakage was observed in the end caps of the secondary electronics case for the water weeped thru welds that sealed up some old screw mounts on the bulkhead end cap. The next part of the test dealt with lowering the vehicle into 100 feet of water and waiting about 10 minutes. When the vehicle was brought up, inspections confirmed the weeping thru the welds on the secondary electronics case. Also, the thruster cases all showed signs of oil leakage. Upon closer inspection, it was determined that all front diaphragms of the thrusters had ruptured and leaked oil. The remainder of the tests were cancelled to make repairs.

All the diaphragms were replaced and thrusters refilled with the non-conducting transformer oil. The endcaps were also repaired by drilling out the weld and replacing it with a threaded screw. Repairs were completed by the end of the first day. Subsequent tests demonstrated that all seals were water tight, as well as the endcap screws and the thruster diaphragms remained intact.

BUOYANCE, STABILITY, AND DRAG CHECKS

Before dynamic vehicle testing could commence, some basic characteristics had to be determined. A theoretical study of the vehicle dynamics was completed in April of 1979 and supplemented by in water test. A brief of this work and some of the calculation tables are presented in Appendix B.

Results of the tests at the lake determined that:

1. With all necessary equipment (ie. batteries, all basic electronics, etc.) the vehicle was 5 lbs. positively buoyant.

2. Vehicle was extremely stable for small weight variations ($\pm$ 5 lbs.).

3. Drag experiments showed that the vehicle had a 5° forward tilt (top is more forward than bottom). This suggests that forward/reverse thrusters are mounted too low. Their horizontal plane should be raised to compensate.

VEHICLE THRUSTER TESTS

These tests were designed to establish the proper operation of the vehicle maneuvering thrusters such as uneven thrust or instability. Relative speeds can be calculated for the various thrusts applied and validated in the tests. A major observation was that the vehicle was deviating from a straight path when thrust was applied from either a forward or reverse direction. The propellers turn in the same direction causing a torque that alters the vehicle path. The higher the prop speeds, the greater the deviation. For example, with the forward thrusters running at full speed, the vehicle swerved from a straight path by about 20° to the left. The solution to the problem would be to use counter-rotating props where the props would create a torque in opposite directions, thereby cancelling each others deviations. This problem not only occurred on the forward/reverse thrusters, but also on the slide and up/down ones as well. The same solution would apply to these as well.

It is noted that this problem can be handled with software modifications. It was further observed that the vehicle moved at a higher velocity in a forward rather than reverse direction. The hydrodynamic design of the thruster casing is seen to allow for the smooth flow of water around it in only one direction. In the opposite direction, the water flow appears to be distorted, diminishing the amount of thrust delivered in the reverse direction.

A third major problem encountered with the thrusters involved the diaphragm seals. which had a tendancy to rip due to the lack of water pressure to compensate for the expansion of the non conductive oil that sealed the thruster motors. This caused many hours to be spent in thruster inspection and maintenance.

Despite these three problem areas, the thrusters supplied adequate thrust and mobility to the EAVE vehicle. They supplied adequate power to keep the vehicle stationary over an object or gave it sufficient thrust to propel and correct the vehicle's speed and direction. Correcting the present problems would increase the power efficiency of the thruster tremendously as well as alleviate some serious control difficulties. These issues do not modify the 1979 pipeline following exercises, but must be considered in on going system design.

MISSION ABORT TIMER (HARDWARE)

The purpose of the hardware abort timers is to shut down power to the vehicle thrusters in the event of a software malfunction. This fail safe capacity would permit recovery of the vehicle in the event of other system failures. Also, when operating without the tether, a mission duration time could be set for the recovery of the vehicle.

The circuitry involved is minimal. It consists of a simple oscillator, a digital counter, and some power switching capability. Before the vehicle is launched, a mission duration time is selected by a rotary switch on the abort timer, and is then packed into the battery pressure case. The on board counter waits till it has counted the preset number of pulses and disables the flow of current from the batteries to the thrusters. The original design of the abort timer called for the use of mechanical relays. However, a byproduct of the vehicle batteries is hydrogen which implied an explosion hazzard. The final design utilized IRF100 High power MOS FET's, which of course would not create sparks. These power transistors delivered enough power to all the thrusters, yet needed very little gate voltage to turn them on or off.

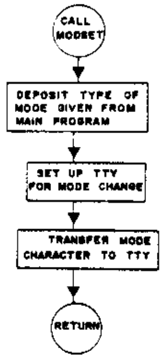
Under actual mission tests, the timer performed excellently. It disabled the vehicle thrusters at preset time intervals of 10, 20, 40, 80, or 120 minutes, which was well in the present mission duration times. Longer times may be required on future missions.

APPENDIX E

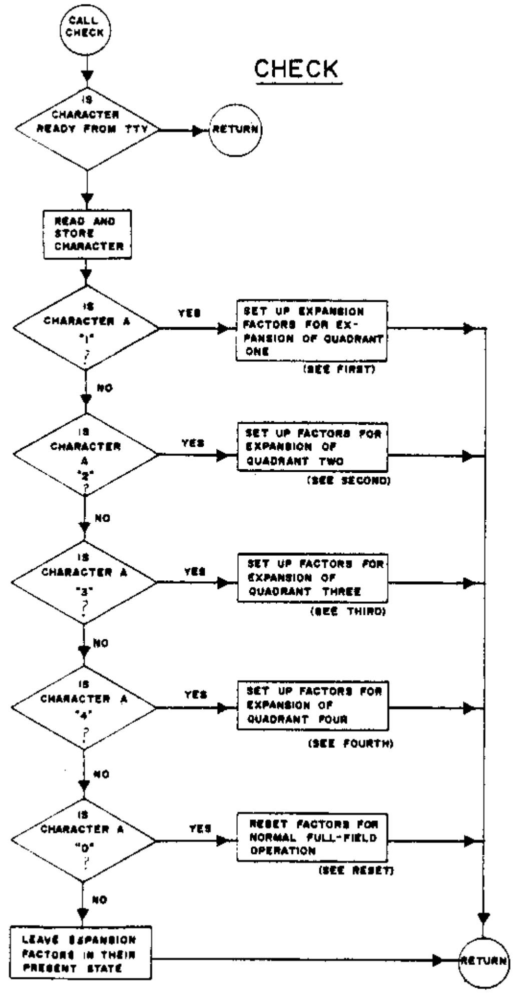
FLOWCHARTS

The following pages include flowcharts for the major routins of the Passive Navigation System software. Only the software which was designed specifically for this system is flowcharted here. None of the floating point software or monitor software (TYPE routine) is included here.

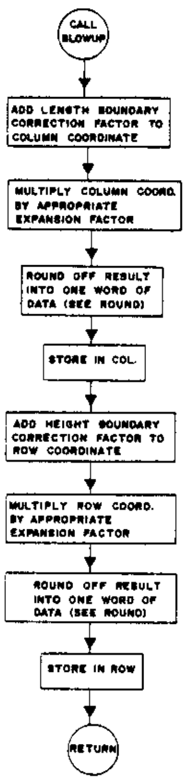
MODSET



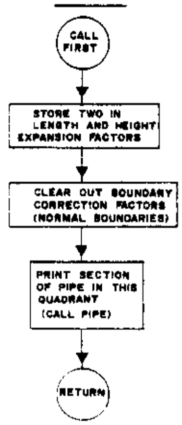
CHECK



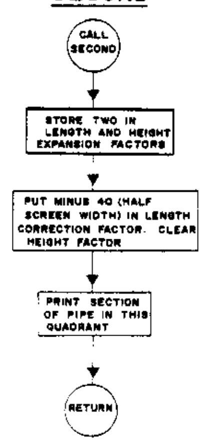
BLOWUP



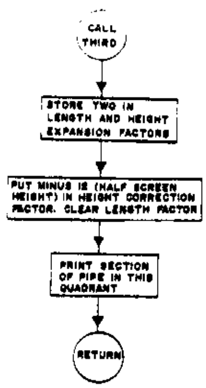
FIRST



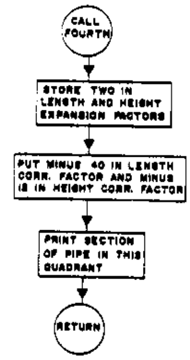
SECOND



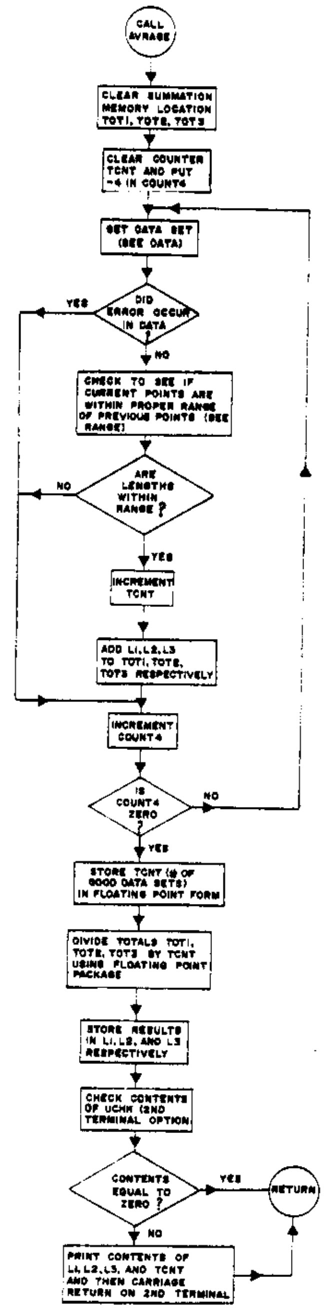
THIRD



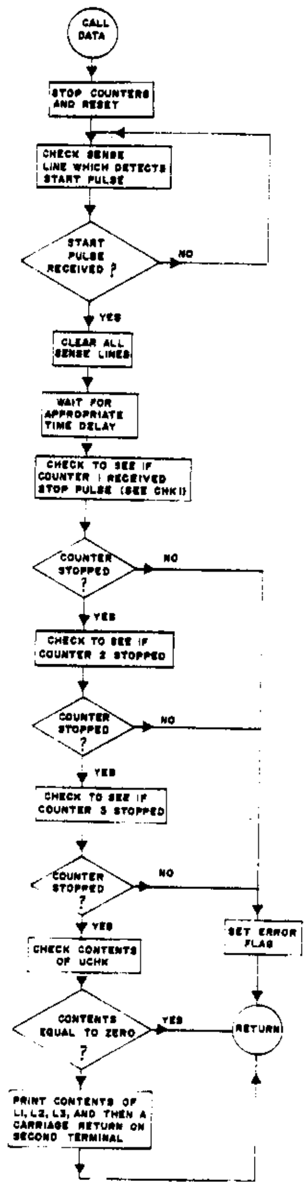
FOURTH



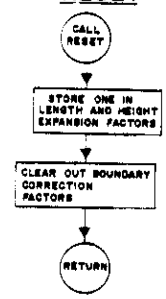
AVRAGE

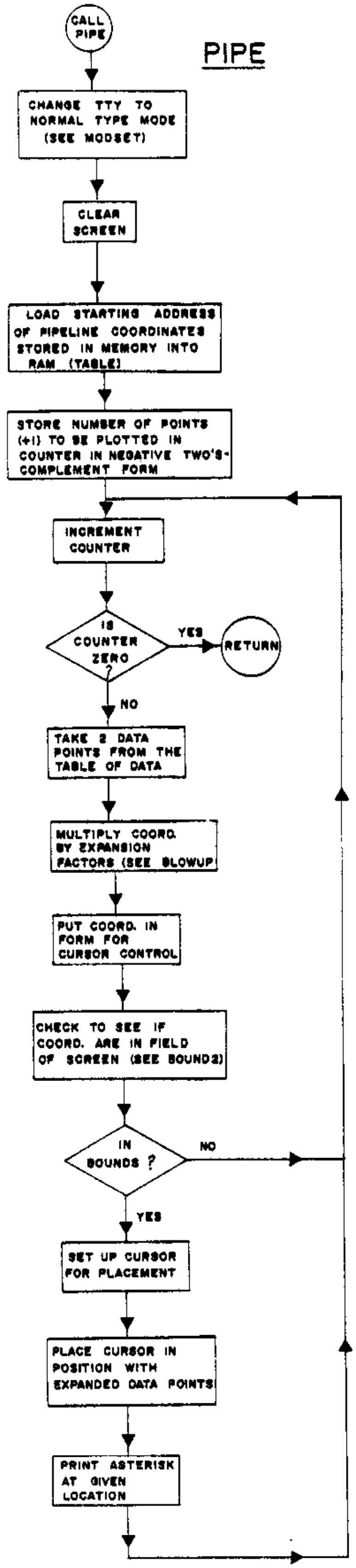


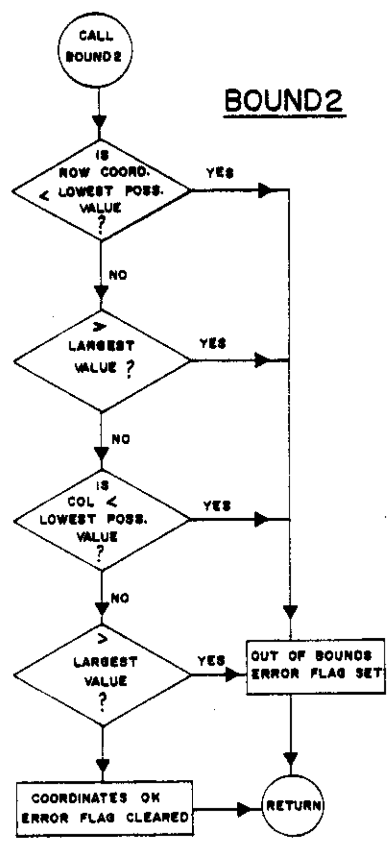
DATA



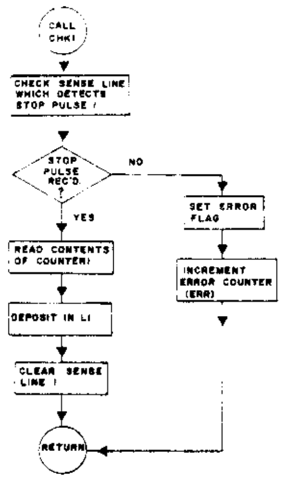
RESET



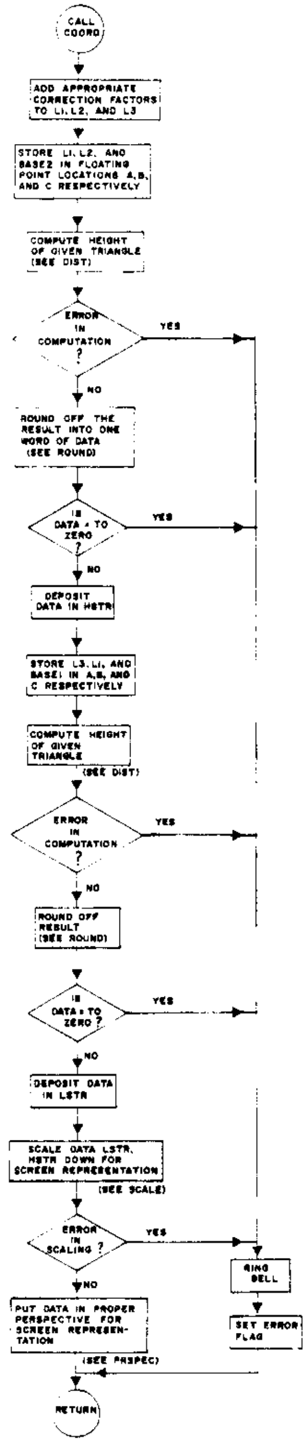

PIPE
CALL PIPE
CHANGE TTY TO NORMAL TYPE MODE (SEE MODSET)
CLEAR SCREEN
LOAD STARTING ADDRESS OF PIPELINE COORDINATES STORED IN MEMORY INTO RAM (TABLE)
STORE NUMBER OF POINTS (+1) TO BE PLOTTED IN COUNTER IN NEGATIVE TWO'S-COMPLEMENT FORM
INCREMENT COUNTER
IS COUNTER ZERO ?
YES
RETURN
NO
TAKE 2 DATA POINTS FROM THE TABLE OF DATA
MULTIPLY COORD. BY EXPANSION FACTORS (SEE BLOWUP)
PUT COORD. IN FORM FOR CURSOR CONTROL
CHECK TO SEE IF COORD. ARE IN FIELD OF SCREEN (SEE BOUND2)
IN BOUNDS ?
NO
YES
SET UP CURSOR FOR PLACEMENT
PLACE CURSOR IN POSITION WITH EXPANDED DATA POINTS
PRINT ASTERISK AT GIVEN LOCATION


BOUND2
CALL BOUND2
IS ROW COORD. < LOWEST POSS. VALUE ?
YES
NO
> LARGEST VALUE ?
YES
NO
IS COL < LOWEST POSS. VALUE ?
YES
NO
> LARGEST VALUE ?
YES
OUT OF BOUNDS ERROR FLAG SET
COORDINATES OK ERROR FLAG CLEARED
RETURN

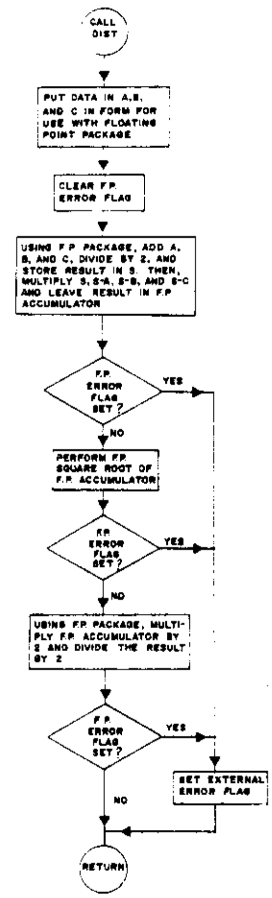
CHK1



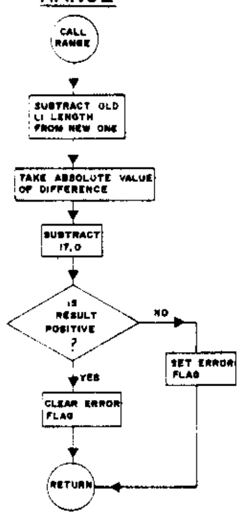
COORD



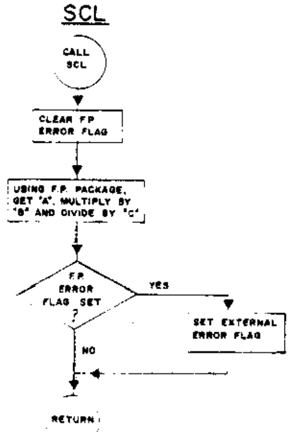
DIST



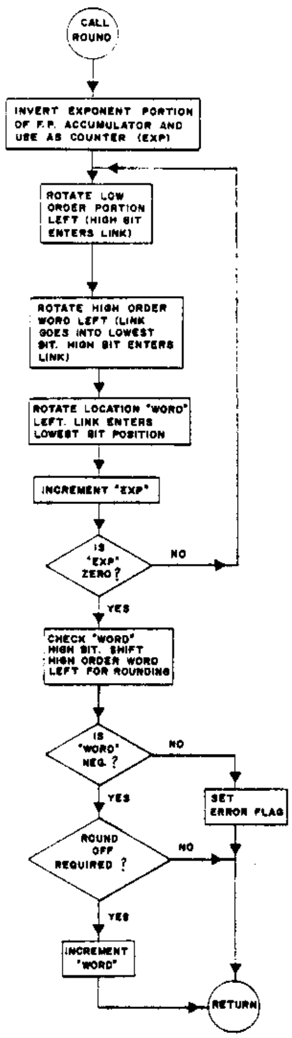
RANGE



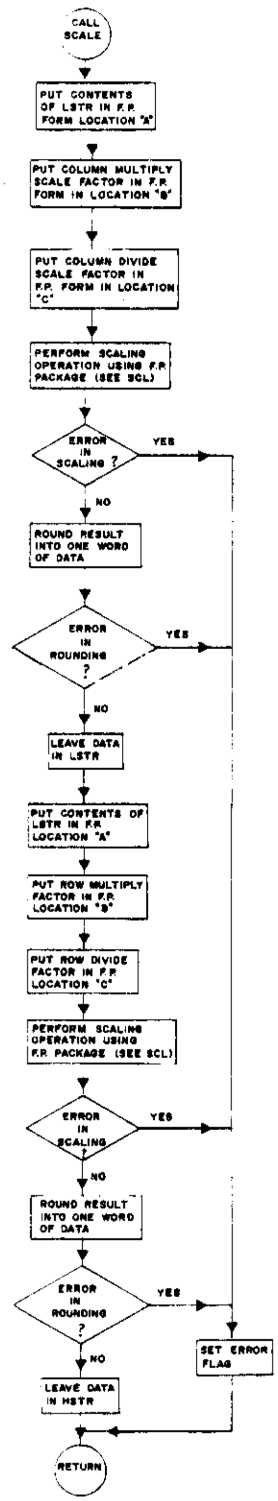
SCL



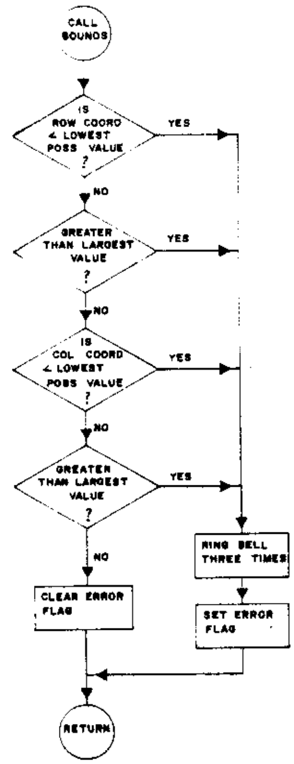
ROUND



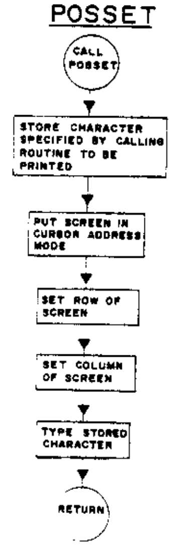
SCALE



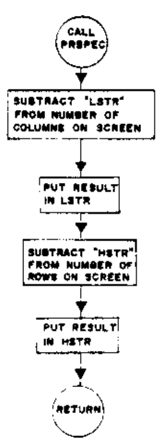
BOUNDS



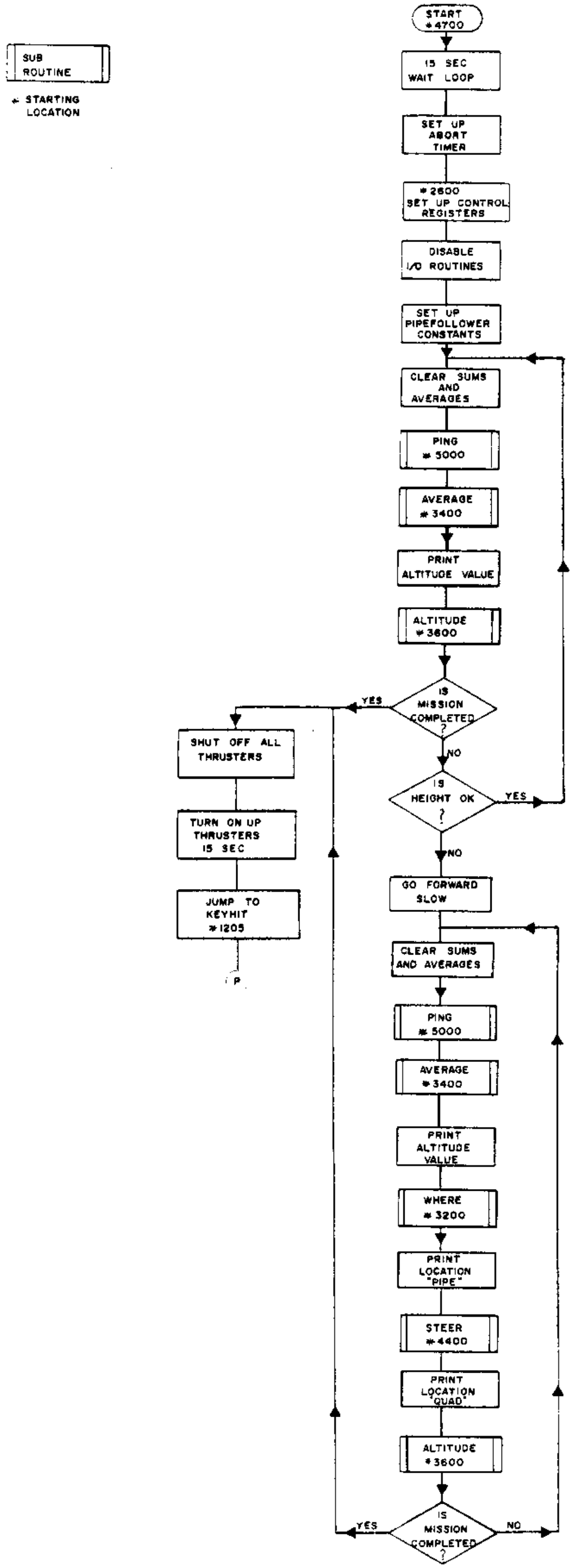
POSSET



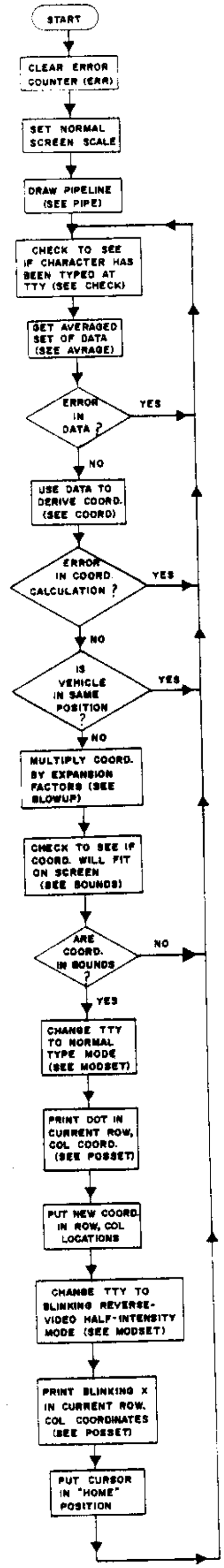
PRSPEC



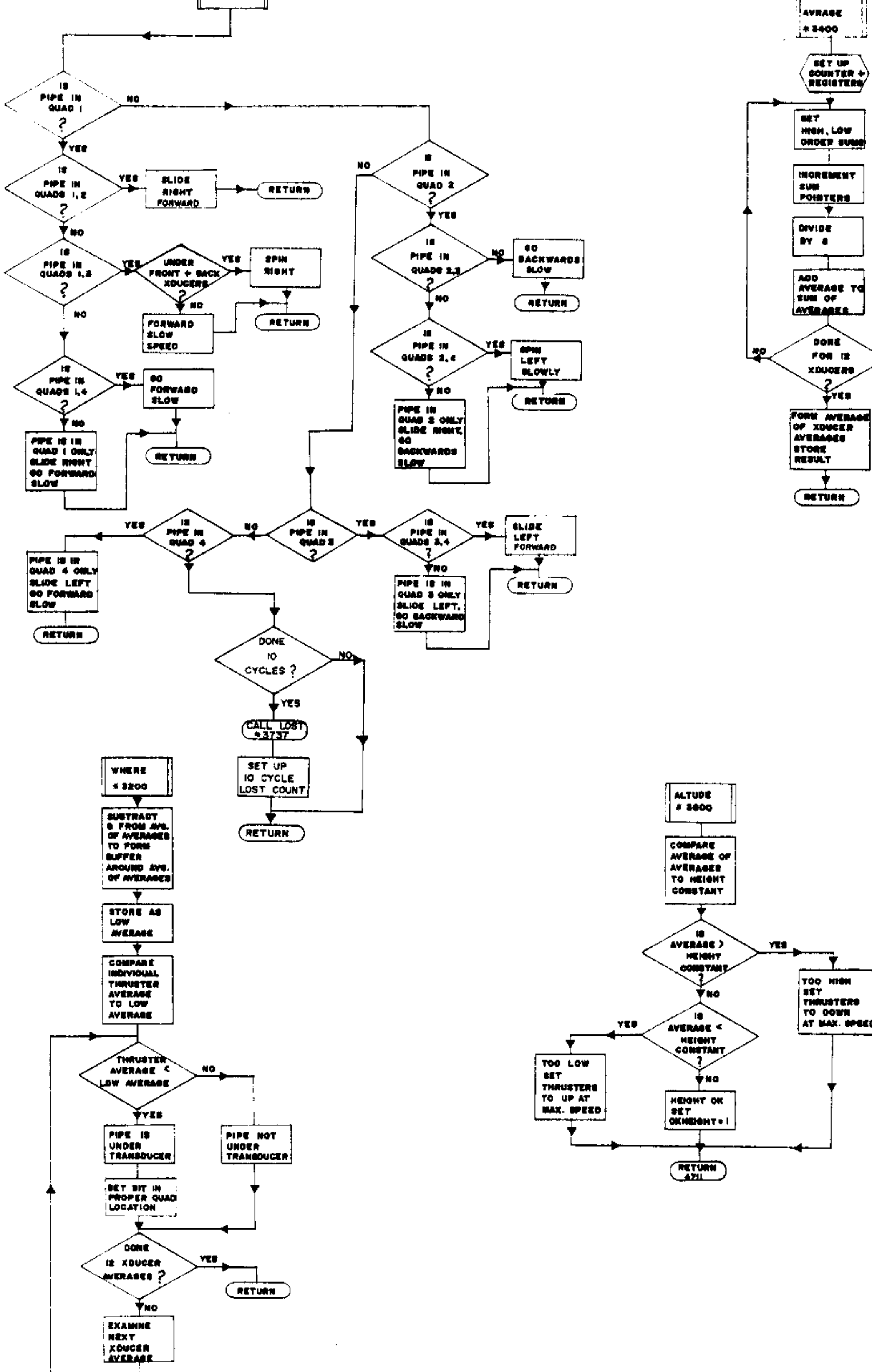
PIPEFOLLOWER
MAIN FLOW

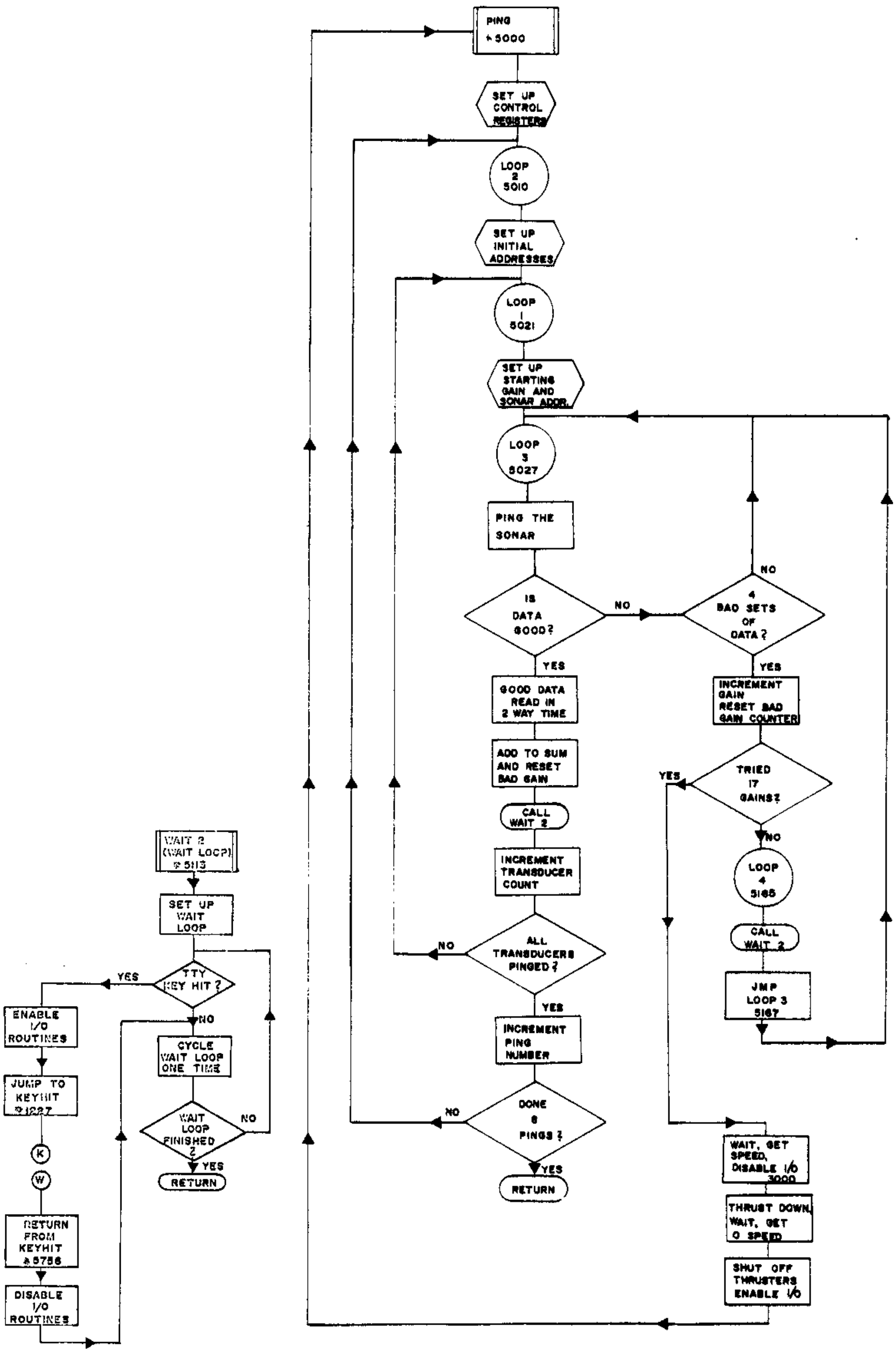


TRACKR

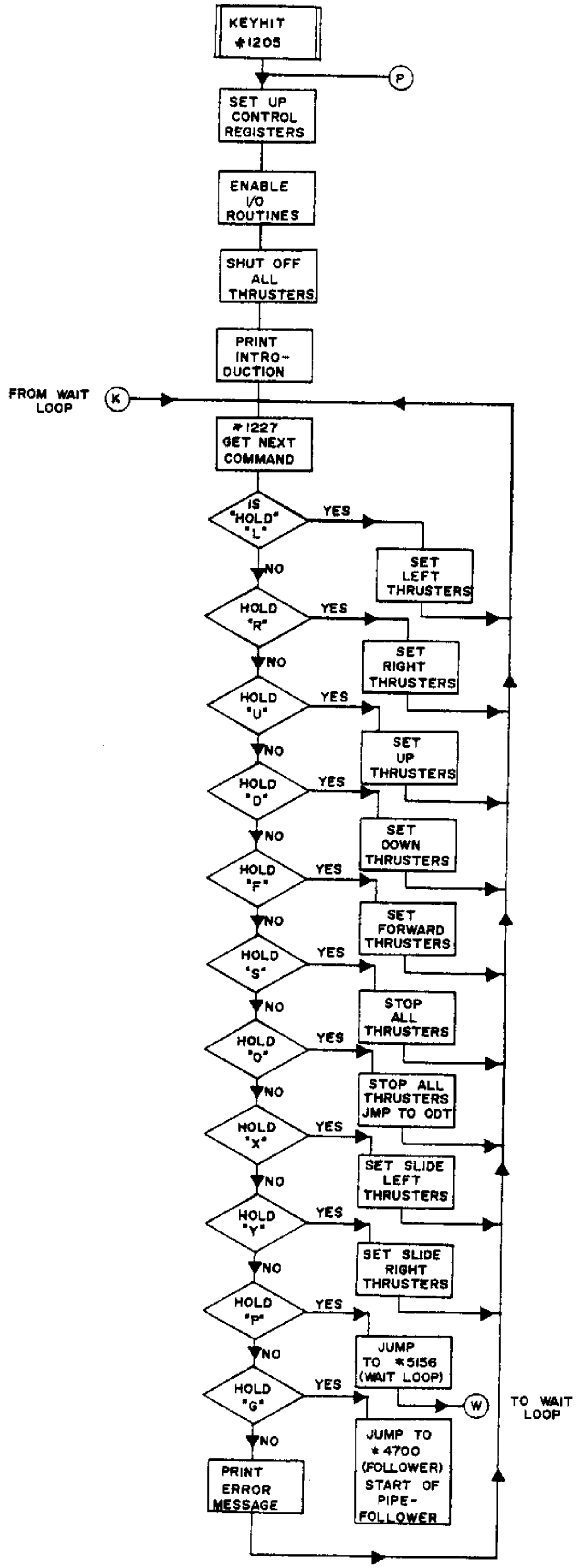

START
CLEAR ERROR COUNTER (ERR)
SET NORMAL SCREEN SCALE
DRAW PIPELINE (SEE PIPE)
CHECK TO SEE IF CHARACTER HAS BEEN TYPED AT TTY (SEE CHECK)
GET AVERAGED SET OF DATA (SEE AVRAGE)
ERROR IN DATA ?
YES
NO
USE DATA TO DERIVE COORD. (SEE COORD)
ERROR IN COORD. CALCULATION ?
YES
NO
IS VEHICLE IN SAME POSITION ?
YES
NO
MULTIPLY COORD. BY EXPANSION FACTORS (SEE BLOWUP)
CHECK TO SEE IF COORD. WILL FIT ON SCREEN (SEE BOUNDS)
ARE COORD. IN BOUNDS ?
NO
YES
CHANGE TTY TO NORMAL TYPE MODE (SEE MODSET)
PRINT DOT IN CURRENT ROW, COL COORD. (SEE POSSET)
PUT NEW COORD. IN ROW, COL LOCATIONS
CHANGE TTY TO BLINKING REVERSE-VIDEO HALF-INTENSITY MODE (SEE MODSET)
PRINT BLINKING X IN CURRENT ROW, COL COORDINATES (SEE POSSET)
PUT CURSOR IN "HOME" POSITION

PIPE FOLLOWER
SUB-ROUTINES

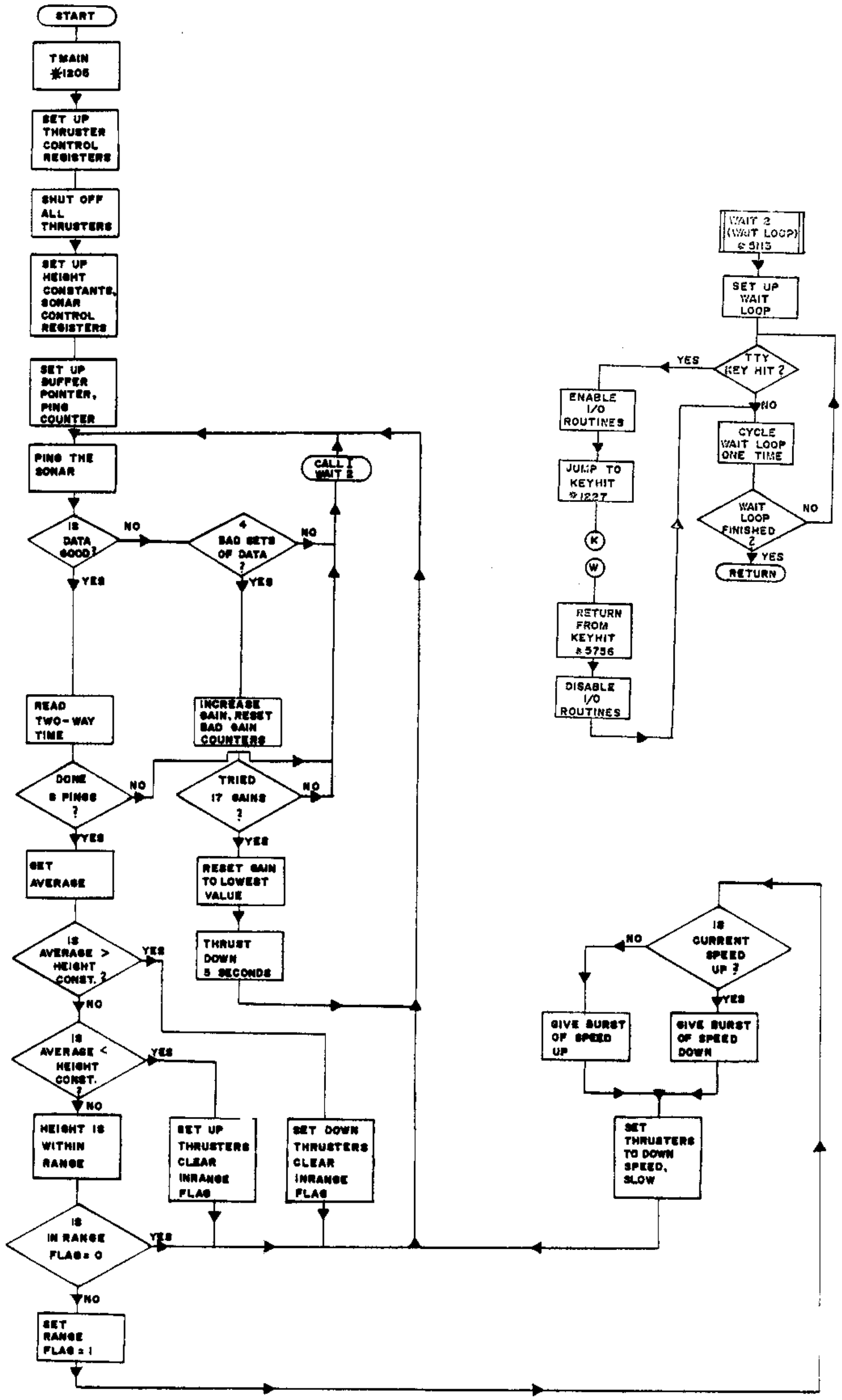




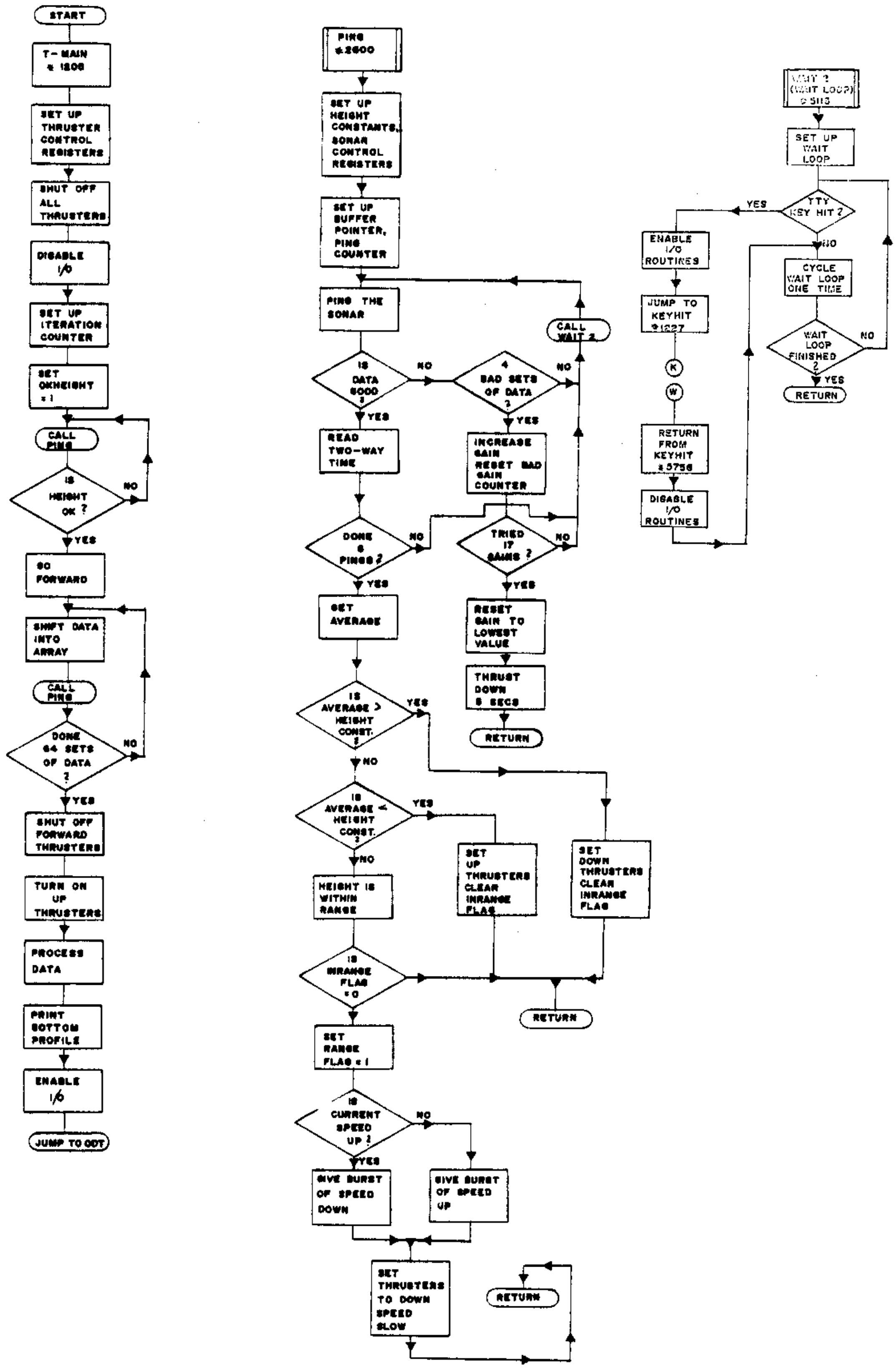
PING
+5000
SET UP CONTROL REGISTERS
LOOP 2 5010
SET UP INITIAL ADDRESSES
LOOP 1 5021
SET UP STARTING GAIN AND SONAR ADDR.
LOOP 3 5027
PING THE SONAR
IS DATA GOOD?
NO
YES
4 BAD SETS OF DATA?
NO
YES
GOOD DATA READ IN 2 WAY TIME
ADD TO SUM AND RESET BAD GAIN
CALL WAIT 2
INCREMENT TRANSDUCER COUNT
ALL TRANSDUCERS PINGED?
NO
YES
INCREMENT PING NUMBER
DONE 8 PINGS?
NO
YES
RETURN
INCREMENT GAIN RESET BAD GAIN COUNTER
TRIED 17 GAINS?
YES
NO
LOOP 4 5165
CALL WAIT 2
JMP LOOP 3 5167
WAIT, GET SPEED, DISABLE I/O 3000
THRUST DOWN, WAIT, GET 0 SPEED
SHUT OFF THRUSTERS ENABLE I/O
WAIT 2 (WAIT LOOP) 5113
SET UP WAIT LOOP
TTY KEY HIT?
YES
NO
ENABLE I/O ROUTINES
JUMP TO KEYHIT 1227
K
W
RETURN FROM KEYHIT 5756
DISABLE I/O ROUTINES
CYCLE WAIT LOOP ONE TIME
WAIT LOOP FINISHED?
NO
YES
RETURN


KEYHIT
#1205
P
SET UP CONTROL REGISTERS
ENABLE I/O ROUTINES
SHUT OFF ALL THRUSTERS
PRINT INTRO-DUCTION
FROM WAIT LOOP
K
#1227 GET NEXT COMMAND
IS "HOLD" "L"
YES
NO
SET LEFT THRUSTERS
HOLD "R"
SET RIGHT THRUSTERS
HOLD "U"
SET UP THRUSTERS
HOLD "D"
SET DOWN THRUSTERS
HOLD "F"
SET FORWARD THRUSTERS
HOLD "S"
STOP ALL THRUSTERS
HOLD "O"
STOP ALL THRUSTERS JMP TO ODT
HOLD "X"
SET SLIDE LEFT THRUSTERS
HOLD "Y"
SET SLIDE RIGHT THRUSTERS
HOLD "P"
JUMP TO *5156 (WAIT LOOP)
W
TO WAIT LOOP
HOLD "G"
JUMP TO *4700 (FOLLOWER) START OF PIPE-FOLLOWER
PRINT ERROR MESSAGE

MANEUVERING PROGRAMS
WITH AUTO ALTITUDE



TRACKLINE PRINTOUT PROGRAMS



APPENDIX F

Navigation System Error Calculation

The Navigation System employs three transducers, at known positions and separations. The mathematics for position calculation in this grid is well established.

It has been noted, however, that if one of the included angles is 90°, the complexity of the positioning computation is reduced significantly. Since this reduces the demand on the microprocessor computer, saves programming time and has no impact on the thrust of the EAVE Program, the simplified algorithm was used.

In practice, however, it is very difficult to emplant buoys precisely in a marine environment. Although leg length can be measured well by acoustic ranging, the angle may not be easily measured. The following analysis reviews the error resulting in position estimate from an error in the included right angle. The mathematics is derived from Figure 16.

The vehicle measures lengths L1, L2, and L3 acoustically, and lengths B1 and B2 are known distances being the system baselines. The included angle has an error, β. The vehicle, assuming it measured L3, in fact measured L3'. The error in L3 is calculated.

From the law of cosines

$$X^2 = 2(B1)^2 \ (1-\cos(\beta)) \tag{1}$$

$$X = (B1) \ (2-2 \ \cos(\beta))^{1/2} \tag{2}$$

$$(L3)^2 = (L3)^2 + X^2 - 2(L3)X \cos(\Omega + 90^\circ - (\beta/2)) \tag{3}$$

$$(L1)^2 = (L3)^2 + (B1)^2 - 2(L3)(B1) \ \cos \Omega \tag{4}$$

Solving (4) for B

$$B = \cos^{-1} \ \frac{(((L3)^2 + (B1)^2 - (L1)^2))}{2(L3)(B1)} \tag{5}$$

By substituting equations (1), (2), and (5) into equation (3), we obtain:

$$\begin{aligned}(L3')^2 = {} & (L3)^2 + 2(B1)^2 \ (1-\cos\beta) \\ & - 2(L3)(B1) \ (2-2\cos(\beta))^{1/2} \\ & \times \cos \ \left(\cos^{-1} \frac{((L3)^2 + (B1)^2 - (L1)^2)}{2(L3)(B1)} + 90^\circ - (\beta/2)\right)\end{aligned}$$

$$\begin{aligned}L3' = {} & ((L3)2 + 2(B1)^2(1-\cos\beta) - 2(L3)(B1) \ (2-2\cos\beta)^{1/2} \ \beta\cos \\ & (\cos^{-1} \frac{((L3)^2 + (B1)^2 - (L1)^2)}{2(L3)(B1)} + 90^\circ - (\beta/2)))^{1/2}\end{aligned}$$

Thus, for a given point inside the operating area, the error in L3 can be determined for a certain angle error. If the general accuracy of the angle can be estimated (eg. $\pm 5^{o}$), then a representative point may be chosen and the error for this point may be found. The error in the plotted point can be found using equations 1, 2, and 3 for L3 and then L3'. (All other parameters are fixed for a given set of data). The percentage error can be found from:

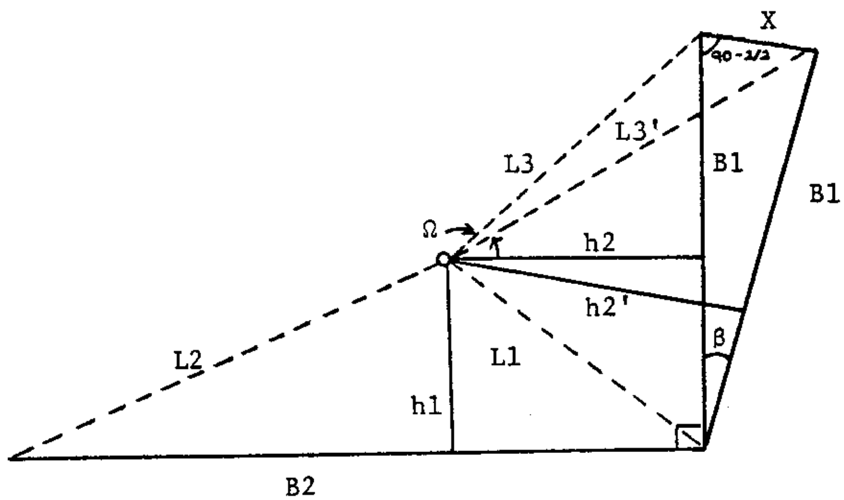
$$\text{percent error} = \frac{h2 - h2'}{h2} \; \beta \; 100 \qquad (6)$$

(Note: Only the h2 component of the coordinates will be affected by this error since it is assumed that the error lies in base B1 and not B2. This affects only L3 and thus, only h2).

Sample calculations indicated that this error will be small for the physical configuration used in testing this system. Thus this configuration is considered acceptable for this purpose.

Figure 16

PHYSICAL CONFIGURATION OF THE SYSTEM WITH ERROR



Physical configuration of the system with error. B1 varies from being at a right angly with B2 by β. The error in computed distance is represented by L3 and the resultant error in calculated coordinate h2 is represented by h2.